



TUGAS AKHIR - KS 141501

**PENGEMBANGAN APLIKASI PETA TIGA DIMENSI JURUSAN
TEKNIK SIPIL INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA MENGGUNAKAN UNITY ENGINE**

***DEVELOPMENT OF 3D INTERACTIVE MAP ON CIVIL
ENGINEERING AT INSTITUT TEKNOLOGI SEPULUH
NOPEMBER USING UNITY ENGINE***

I Gst Bagus Rogeri A.
NRP 5211100121

Dosen Pembimbing
Dr. Eng. Febriliyan Samopa, S. Kom, M. Kom
Nisfu Asrul Sani, S. Kom, M. Sc

JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017



TUGAS AKHIR- KS 091336

**PENGEMBANGAN APLIKASI PETA
INTERAKTIF TIGA DIMENSI JURUSAN
TEKNIK SIPIL INSTITUT TEKNOLOGI
SEPULUH NOPEMBER SURABAYA
MENGUNAKAN UNITY ENGINE**

**I GST BAGUS ROGERI A.
NRP 5211 100 121**

**Dosen Pembimbing I
Dr. Eng. Febriliyan Samopa, S. Kom, M. Kom**

**Dosen Pembimbing II
Nisfu Asrul Sani, S. Kom, M. Sc**

**JURUSAN SISTEM INFORMASI
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember
Surabaya 2017**



FINAL PROJECT - KS 091336

**DEVELOPMENT OF 3D INTERACTIVE MAP
ON CIVIL ENGINEERING AT INSTITUT
TEKNOLOGI SEPULUH NOPEMBER USING
UNITY ENGINE**

**I GST BAGUS ROGERI A.
NRP 5211 100 121**

**Supervisor I
Dr. Eng. Febriliyan Samopa, S. Kom, M. Kom**

**Supervisor II
Nisfu Asrul Sani, S. Kom, M. Sc**

**INFORMATION SYSTEMS DEPARTMENT
Information Technology Faculty
Institut Teknologi Sepuluh Nopember
Surabaya 2017**

LEMBAR PENGESAHAN

PENGEMBANGAN APLIKASI PETA INTERAKTIF TIGA DIMENSI PADA JURUSAN TEKNIK SIPIL INSTITUT TEKNOLOGI SEPULUH NOPEMBER MENGGUNAKAN UNTIY ENGINE

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Departemen Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

IGST BAGUS ROGERIA.
NRP 5211 100 121

Surabaya, Juli 2017

**KETUA
JURUSAN SISTEM INFORMASI**



Dr. Ir. Aris Tjahyanto, M.Kom.
NIP.19650310 199102 1 001

LEMBAR PERSETUJUAN
PENGEMBANGAN APLIKASI PETA
INTERAKTIF TIGA DIMENSI JURUSAN TEKNIK
SIPIL INSTITUT TEKNOLOGI SEPULUH
NOPEMBER SURABAYA MENGGUNAKAN
UNITY ENGINE

TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Komputer
pada
Jurusan Sistem Informasi
Fakultas Teknologi Informasi
Institut Teknologi Sepuluh Nopember

Oleh :

I GST BAGUS ROGERI A.
NRP 5211 100 121

Disetujui Tim Penguji : Tanggal Ujian : 5 Juli 2017
Periode Wisuda : September 2017

Dr. Eng. Febriliyan Samopa, S. Kom, M. Kom (Pembimbing I)

Nisfu Asrul Sani, S. Kom, M.Sc

(Pembimbing II)

Dr. Ir. Aris Tjahyanto, M. Kom

(Penguji I)

Edwin Riksakomara, S. Kom, MT.

(Penguji II)

PENGEMBANGAN APLIKASI PETA INTERAKTIF TIGA DIMENSI PADA JURUSAN TEKNIK SIPIL INSTITUT TEKNOLOGI SEPULUH NOPEMBER MENGGUNAKAN UNITY ENGINE

Nama Mahasiswa : I Gst Bagus Rogeri A.
NRP : 5211 100 121
Jurusan : Sistem Informasi FTIF-ITS
Dosen Pembimbing I : Dr. Eng. Febriliyan Samopa, S.
Kom., M. Kom.
Dosen Pembimbing II : Nisfu Asrul Sani, S. Kom, M. Sc

ABSTRAK

Era teknologi informasi saat ini sudah memungkinkan kita untuk dapat melakukan suatu proses secara virtual, seperti salah satunya adalah dalam bentuk peta interaktif tiga dimensi, dimana proses untuk mengunjungi suatu tempat digantikan dengan mengunjungi suatu dunia virtual dalam bentuk peta interaktif tiga dimensi, dimana proses untuk mengunjungi suatu tempat digantikan dengan mengunjungi suatu dunia virtual dalam bentuk peta tiga dimensi dari lokasi tersebut. Proses virtualisasi semacam ini yang diharapkan untuk dapat membantu penggunaannya mendapatkan informasi secara lebih menarik. Peta interaktif tiga dimensi semacam ini merupakan salah satu topik penelitian mahasiswa ITS terbukti dengan beberapa judul tugas akhir yang mengambil tema peta tiga dimensi dengan objek lokasi yang berbeda-beda. Objek dalam penelitian ini, Jurusan Teknik Sipil ITS, sudah pernah menjadi objek penelitian sebelumnya mengenai peta interaktif tiga dimensi dengan menggunakan Unreal Engine, namun setelah dibuatkan, dari pihak klien masih merasa tidak tahu bagaimana cara memakainya dan melakukan perubahan pada aplikasi tersebut. Dari penelitian sebelumnya, hal tersebut juga disarankan untuk dapat mengembangkan peta interaktif tiga

dimensi tersebut supaya dapat digunakan secara online pada web, sehingga dapat salah satu keputusannya adalah untuk menggunakan Unity Engine yang mampu untuk membuat game untuk ditaruh pada browser. Pada penelitian tentang Unity sebelumnya juga tidak terlihat penggunaan database pada hosting, yang mana apabila bisa dilakukan, akan dapat mempermudah perubahan isi konten seperti tulisan, gambar, dan video.

Oleh karena itu, penulis akan mengembangkan peta interaktif tiga dimensi dari Jurusan Teknik Sipil ITS dengan menggunakan Unity Engine sehingga memungkinkan untuk membuat virtual reality tersebut dapat dijalankan di dalam sebuah web, dan menghubungkan aplikasi dengan database dalam sebuah web hosting.

Kata Kunci: 3D, Unity, Peta Interaktif, Teknik Sipil ITS.

***DEVELOPMENT OF 3D INTERACTIVE MAP ON CIVIL
ENGINEERING AT INSTITUT TEKNOLOGI SEPULUH
NOPEMBER SURABAYA USING UNTIY ENGINE***

Student Name : I Gst Bagus Rogeri A.
NRP : 5211 100 121
Department : Sistem Informasi FTIF-ITS
Supervisor I : Dr, Eng. Febriliyan Samopa, S.
Kom, M. Kom
Supervisor II : Nisfu Asrul Sani, S. Kom, M. Sc

ABSTRACT

The era of information technology (IT) is now required to perform a virtual process, such as one in the form of three-dimensional interactive map, where the process for a place is replaced with a virtual world in the form of three-dimensional interactive map, where the process for a place is replaced with a virtual world in the form of a three-dimensional map of the intended location. This kind of virtualization process is expected to help further information. This three-dimensional interactive map is one of ITS student's research topics that is proven by some final project titles that take this theme of three-dimensional map using Unreal Engine, but after those are being made, the client doesn't seem to know how to use and update those programs. From previous researches, it is also suggested that the next research to be able to develop three-dimensional interactive map that could be used online on the web. Previous researches that uses Unreal Engine or Unity Engine doesn't seem to have access database from web hosting, whereas ever since Unity5 comes out, those feature for accessing web databases are made possible even for non-paying user, where we could take value of text, images, videos, or some Unity objects other than scripts from the web that is compressed with Unity's AssetBundles.

Therefore, the author will develop three-dimensional interactive maps of ITS' Civil Engineering Department by using Unity Engine so it would be possible to create virtual reality as it can be run and organized in a dynamic web simulation, and making the program to be able to access data from the web.

Keywords: 3D, Unity, Interactive Map, Civil Engineering, ITS

KATA PENGANTAR

Segala puji bagi Allah SWT yang telah memberikan rahmat-Nya kepada penulis hingga dapat menyelesaikan Tugas Akhir ini yang merupakan salah satu syarat kelulusan di Jurusan Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Tak lupa penulis juga mengucapkan terima kasih sebanyak – banyaknya kepada semua pihak yang sudah berkenan membantu, terutama kepada:

- Bapak Febriliyan Samopa selaku pembimbing I dalam proses pengerjaan tugas akhir ini yang dengan sabar membimbing pengerjaan Tugas Akhir hingga selesai.
- Bapak Nisfu Asrul Sani selaku pembimbing II dalam membantu untuk meningkatkan motivasi menjalankan proses pengerjaan tugas akhir ini hingga selesai.
- Orang tua, saudara, dan teman-teman yang membantu dari meningkatkan motivasi hingga membantu menjelaskan tata acara penyelesaian tugas akhir.

Penulis tetap menyadari bahwa tugas akhir ini masih memiliki banyak kekurangan dan ketidaksempurnaan, untuk itu penulis mengharapkan saran atas tugas akhir ini yang bersifat membangun guna perbaikan di masa mendatang. Akhir kata, penulis berharap tugas akhir ini dapat bermanfaat bagi kita semua.

Surabaya, Juli 2017

Penulis

Halaman ini sengaja dikosongkan

DAFTAR ISI

ABSTRAK	i
ABSTRACT	iii
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xiv
DAFTAR POTONGAN KODE	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Rumusan Masalah	3
1.3 Batasan Masalah	3
1.4 Tujuan Tugas Akhir	4
1.5 Manfaat Tugas Akhir	4
1.6 Relevansi	5
BAB II TINJAUAN PUSTAKA	7
2.1 Studi Sebelumnya	7
2.2 Dasar Teori	8
2.2.1 Peta Tiga Dimensi	8
2.2.2 Game Engine	9
2.2.3 Unity Game Engine	9
2.2.4 Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember	10
2.2.5 Aplikasi Modeling 3D	12
2.2.6 Program Pengolahan Gambar	12
2.2.7 Agile ICONIX Practice	13
BAB III METODOLOGI	17
3.1 Tahapan Pelaksanaan Tugas Akhir	17
3.1.1 Studi Literatur	18
3.1.2 Studi Kebutuhan	18
3.1.3 Desain Sistem	19
3.1.4 Pembuatan Aplikasi	20
3.1.5 Uji Coba Sistem Aplikasi	21
3.1.6 Penulisan Laporan	21

BAB IV PERANCANGAN	23
4.1 Analisis Kebutuhan Sistem	23
4.1.1 Kebutuhan Fungsional Sistem	23
4.1.2 Kebutuhan Non-Fungsional Sistem	23
4.1.3 Analisis Pemilihan Tombol & Navigasi.....	24
4.1.4 Daftar Aktor.....	26
4.2 Desain Sistem	27
4.2.1 Pembuatan GUI Storyboard	27
4.2.2 Pembuatan Domain Model	29
4.2.3 Pembuatan Use Case	30
4.2.4 Pembuatan Robustness Analysis	33
4.2.5 Penggunaan Sequence Diagram.....	33
4.2.6 Penggunaan Test Case.....	33
BAB V IMPLEMENTASI DAN UJICOBAN	35
5.1 Lingkungan Implementasi.....	35
5.2 Implementasi Aplikasi	36
5.2.1 Model 3D Gedung Jurusan Teknik Sipil ITS	36
5.2.2 Peta Dua Dimensi (2D)	38
5.2.3 Pembuatan Database	40
5.2.4 Pembuatan Halaman PHP	42
5.2.5 Pembuatan Aset Aplikasi	50
5.2.6 Pencahayaan	76
5.2.7 Pengaturan Akhir	77
5.3 Uji Coba Sistem.....	77
5.3.1 Uji Coba Fungsional	79
5.3.2 Uji Coba Non-Fungsional	79
BAB VI HASIL PEMBAHASAN UJI COBAN	83
6.1 Hasil Uji Coba Fungsionalitas.....	83
6.2 Hasil Uji Coba Non-Fungsional	84
6.2.1 Hasil Uji Kemampuan Aplikasi saat Koneksi dengan Host Terputus	84
6.2.2 Hasil Uji Kemampuan Aplikasi saat Diakses oleh Lebih dari Satu Pengguna	84
6.2.3 Hasil Uji Kemampuan Aplikasi Ketika Ada Perubahan dalam Database.....	85
6.2.4 Hasil Uji Kemampuan Aplikasi untuk Menerima Video dengan Ukuran Besar	85

6.2.5 Hasil Uji Performa, Uji Platform Web, Uji Kompatibilitas Browser, dan Uji Kompatibilitas Sistem Operasi	85
6.2.6 Hasil Evaluasi Implementasi 3D	89
BAB VII KESIMPULAN DAN SARAN	91
7.1 Kesimpulan	91
7.2 Saran	92
DAFTAR PUSTAKA	93
BIODATA PENULIS	95
LAMPIRAN A USE CASE	1
Use Case Descriptions	1
LAMPIRAN B ROBUSTNESS DIAGRAM	1
LAMPIRAN C SEQUENCE DIAGRAM	1
LAMPIRAN D TEST CASE	1
LAMPIRAN E DENAH BANGUNAN	1

Halaman ini sengaja dikosongkan

DAFTAR GAMBAR

Gambar 2-1 Agile ICONIX Practice.....	13
Gambar 3-1 Flow Chart tahap pengerjaan tugas akhir.....	17
Gambar 3-2 ICONIX Process.....	19
Gambar 4-1 Prototipe Menu Utama.....	28
Gambar 4-2 Prototipe Menu About	28
Gambar 4-3 Prototipe pause menu.....	29
Gambar 4-4 Prototipe daftar video laboratorium.....	29
Gambar 4-5 Domain Model.....	30
Gambar 4-6 Use Case Model untuk Pengguna Umum	32
Gambar 4-7 Use Case Model untuk Administrator	32
Gambar 5-1 Model 3D Gedung Jurusan Teknik Sipil ITS.....	36
Gambar 5-2 Tata Letak Gedung Teknik Sipil ITS.....	39
Gambar 5-3 Foto Bangunan Gedung Teknik Sipil ITS diambil dengan drone	39
Gambar 5-4 Visualisasi Foreign Key antara video_lab dengan laboratorium_sipil.....	40
Gambar 5-5 Tampilan halaman PHP upload_videoLab.php.....	44
Gambar 5-6 Membuat Denah menggunakan Blender.....	50
Gambar 5-7 Meninggikan Denah sebagai Dinding menggunakan fitur Extrude pada Blender.....	51
Gambar 5-8 Buat Material danbaru	52
Gambar 5-9 Buat Tekstur Baru.....	52
Gambar 5-10 (Kiri) Edit Mode dengan Viewport Texture dan (Kanan) UV/Image Editor.....	53
Gambar 5-11 Pengaturan Material dan Tekstur pada Unity.....	53
Gambar 5-12 Prefab untuk Mesin.....	55
Gambar 5-13 Penanda Objek.....	55
Gambar 5-14 Prefab FPSController.....	56
Gambar 5-15 Public Value yang dapat dibuat diubah pada FPSController dari StandardAsset	56
Gambar 5-16 Pengaturan Kualitas Unity	57
Gambar 5-17 Tampilan kualitas Fast.....	57
Gambar 5-18 Tampilan Kualitas Good.....	58
Gambar 5-19 Tampilan Kualitas Fantastic.....	58
Gambar 5-20 Menu Utama Aplikasi.....	60

Gambar 5-21 Pause Menu	60
Gambar 5-22 Antarmuka Daftar Laboratorium.....	61
Gambar 5-23 Trigger Area untuk Laboratorium Hidroteknik	63
Gambar 5-24 Objek yang akan Menjadi Prefab bagi listVideoLab	66
Gambar 5-25 Tampilan Antarmuka untuk Prefab (Kiri) dan Komponen Vertical Layout Group (Kanan).....	67
Gambar 5-26 Tampilan RawImage sebagai Video Player	70
Gambar 5-27 Pembuatan Animasi dalam Blender	71
Gambar 5-28 Komponen Objek blend dengan Animasi saat Ditaruh Dalam Scene	72
Gambar 5-29 Ganti Animation Type	73
Gambar 5-30 Inspector dari Objek blend.....	73
Gambar 5-31 Tampilan Mini-Map	75
Gambar 5-32 Komponen dari Kamera Mini-Map	76
Gambar 11-1 Robustness Diagram untuk Menu About, Puase Menu, dan Navigasi	1
Gambar 11-2 Robustness Diagram untuk Pengguna Umum kepada Mesin dan Animasinya.....	2
Gambar 11-3 Robustness Diagram untuk Pengguna Umum terhadap Daftar Video	2
Gambar 11-4 Robustness Diagram untui Administrator dalam Mengatur Database	2
Gambar 12-1 Sequence Diagram untuk Membuka Menu Utama, Menu About, dan Pause Menu	1
Gambar 12-2 Sequence Diagram untuk Menjalankan Simulasi Mesin	1
Gambar 12-3 Sequence Diagram untuk Navigasi Karakter	2
Gambar 12-4 Sequence Diagram untuk Menampilkan Daftar Video dan Memainkan Video.....	2
Gambar 12-5 Sequence Diagram untuk Administrator Melakukan Perubahan Data	3
Gambar 14-1 Denah Gedung A Tampak 3 dan Gedung C Tampak 4	1
Gambar 14-2 Denah Gedung B Tampak 1 dan 2	2
Gambar 14-3 Denah Gedung C lantai 1, 2, dan 3.....	3
Gambar 14-4 Denah Gedung E lantai 1 dan 2.....	4

Gambar 14-5 Denah Gedung E tampak 3 dan 4	5
Gambar 14-6 Denah Gedung E Tampak Potongan.....	6
Gambar 14-7 Denah Gedung H Lantai 1 dan 2	7
Gambar 14-8 Denah Gedung H Tampak 1, 2, 3, dan 4.....	8
Gambar 14-9 Gedung I Lantai 1 dan 2.....	9
Gambar 14-10 Denah Gedung I Tampak 1, 2, 3, dan 4	10
Gambar 14-11 Denah Gedung J Lantai 1 dan 2.....	11
Gambar 14-12 Denah Gedung J Tampak 1, 2, 3, dan 4	12
Gambar 14-13 Denah Gedung L Lantai 1, Lantai 1 1/2, Tampak 1, dan Tampak 2	13
Gambar 14-14 Denah Gedung L Tampak Potongan.....	14

Halaman ini sengaja dikosongkan

DAFTAR TABEL

Tabel 4-1 Daftar Kebutuhan Fungsional Sistem.....	24
Tabel 4-2 Daftar Kebutuhan Non-Fungsional Sistem.....	25
Tabel 4-3 Analisis Pemilihan Tombol Navigasi.....	25
Tabel 4-4 Daftar Aktor.....	27
Tabel 4-5 Pemetaan Use Case berdasarkan kebutuhan fungsional.....	31
Tabel 5-1 Spesifikasi Komputer I.....	35
Tabel 5-2 Spesifikasi Komputer II.....	36
Tabel 5-3 Daftar Test Case beserta Use Case dan Kebutuhan Fungsional maupun Non-Fungsional Terkait.....	78
Tabel 6-1 Hasil dari test case.....	83
Tabel 6-2 Spesifikasi Perangkat Pengujian.....	86
Tabel 6-3 Loadtime dan Rerata Frame per Second.....	87
Tabel 6-4 Kompatibilitas Browser.....	88
Tabel 6-5 Implementasi 3D.....	89
Tabel 10-1 Use Case Description untuk UC-101 Pengguna Umum Membuka Menu Utama.....	1
Tabel 10-2 Use Case Description Untuk Pengguna Umum Membuka Menu About.....	2
Tabel 10-3 Use Case Description untuk Pengguna Umum melakukan Navigasi dalam Peta 3D.....	2
Tabel 10-4 Use Case Description Pengguna Umum Menjalankan Simulasi.....	4
Tabel 10-5 Use Case Description Pengguna Umum Menampilkan daftar Video dari Laboratorium.....	5
Tabel 10-6 Use Case Description Pengguna Umum Mendapatkan Daftar Video Berdasarkan Database.....	6
Tabel 10-7 Use Case Description Pengguna Umum Menampilkan Video.....	7
Tabel 10-8 Use Case Description Pengguna Umum Mendapatkan Video dari Database.....	8
Tabel 10-9 Use Case Description Pengguna Umum Menampilkan Menu Pause.....	9
Tabel 10-10 Use Case Description Pengguna Umum Menampilkan Mini-map.....	10

Tabel 10-11 Use Case Description Administrator mengakses halaman pengaturan web.....	11
Tabel 10-12 Use Case Description Administrator Melakukan Update atau Upload Teks pada Database.....	12
Tabel 10-13 Use Case Description Administrator Melakukan Update atau Upload Video	13
Tabel 13-1 TC-01: Membuka aplikasi dalam host	1
Tabel 13-2 TC-02: Membuka aplikasi diluar host.....	2
Tabel 13-3 TC-03: Menampilkan menu about dengan koneksi host.....	3
Tabel 13-4 TC-04: Menampilkan menu about tanpa koneksi host.....	4
Tabel 13-5 TC-05: Pengguna menjelajahi peta interaktif	5
Tabel 13-6 TC-06: Pengguna menjelajahi peta interaktif tanpa koneksi host.....	7
Tabel 13-7 TC-07: Menjalankan simulasi mesin.....	8
Tabel 13-8 TC-08: Menjalankan simulasi mesin tanpa koneksi host.....	9
Tabel 13-9 TC-09: Uji Tampilkan Daftar Video berdasarkan Database.....	10
Tabel 13-10 Uji memanggil daftar video tanpa koneksi host.	11
Tabel 13-11 TC-11: Uji menampilkan video	12
Tabel 13-12 TC-12: Uji pause menu saat melakukan navigasi	13
Tabel 13-13 TC13: Uji mini-map	14
Tabel 13-14 TC-14: Uji unggah video dengan berbagai ukuran	15
Tabel 13-15 TC-15: Uji membuka aplikasi secara bersamaan	16
Tabel 13-16 TC-16: Uji membuka video secara bersamaan ..	17

DAFTAR POTONGAN KODE

Potongan Kode 5-1 Query untuk membuat `fullview_video_lab`	41
Potongan Kode 5-2 Query untuk membuat `no64_videoLab`	41
Potongan Kode 5-3 upload_videoLab.php melakukan koneksi ke database	42
Potongan Kode 5-4 upload_videoLab.php mengambil daftar laboratorium dan daftar video	43
Potongan Kode 5-5 Menaruh Daftar Laboratorium dan Video pada Dropdown	44
Potongan Kode 5-6 upload_videoLab.php memeriksa file unggahan dan kode memperbarui video dalam database	45
Potongan Kode 5-7 Kode upload_videolab.php untuk mengunggah video baru ke dalam database	46
Potongan Kode 5-8 Source Code untuk json_videolist.php	47
Potongan Kode 5-9 Kode PHP untuk display_videoLab.php	49
Potongan Kode 5-10 mainMenu.cs	59
Potongan Kode 5-11 pauseScreen.cs	61
Potongan Kode 5-12 showCanvas_listVideo.cs	63
Potongan Kode 5-13 listVideoLab.cs bagian 1: Variabel dan Start()	65
Potongan Kode 5-14 listVideoLab.cs bagian 2: Update() dan ambilJson()	66
Potongan Kode 5-15 buttonPlayVidLab.cs	68
Potongan Kode 5-16 videoDisplayer.cs Bagian 1: Variabel, Update(), dan mulai()	69
Potongan Kode 5-17 videoPlayer.cs Bagian 2: playVid()	70
Potongan Kode 5-18 Variabel Objek dan Animasi	74
Potongan Kode 5-19 Pengaturan Isi Variabel untuk Animasi	74
Potongan Kode 5-20 Menjalankan Animasi Default	74
Potongan Kode 5-21 Menjalankan Animasi Berdasarkan Potongan Animasi	74
Potongan Kode 5-22 Script untuk Mengikuti Player	75

Halaman ini sengaja dikosongkan

BAB I

PENDAHULUAN

Pada bab ini, akan dijelaskan tentang Latar Belakang Masalah, Perumusan Masalah, Batasan Masalah, Tujuan Tugas Akhir, Relevansi dan Manfaat Kegiatan Tugas Akhir.

1.1 Latar Belakang Masalah

Teknologi pada saat ini sudah banyak yang berkembang. Salah satu jenisnya adalah bertambahnya masyarakat virtual (*virtual society*), dimana beberapa proses yang sebelumnya dijalankan secara tradisional, yakni secara fisik, sekarang sudah mulai dilakukan secara elektronik atau cara virtual lain [1]. Sudah mulai ada beberapa proses yang dibuat versi virtualnya, salah satunya seperti sosial media yang dalam bentuk seperti *facebook*, *twitter*, dan sebagainya, yang merupakan virtualisasi proses dari berkomunikasi dengan teman dan *online shopping* yang merupakan virtualisasi dari proses berbelanja di toko. Kasus yang ingin diangkat dalam penelitian ini adalah bagaimana dapat membuat virtualisasi proses dari seseorang yang dapat mengunjungi suatu tempat tanpa perlu ke tempat tersebut, dan melakukan interaksi di sana untuk mempelajari informasi lebih lanjut. Untuk melakukan virtualisasi dari proses tersebut, dalam penelitian ini akan digunakan konsep *Virtual Reality*.

Virtual Reality (VR) atau Realitas Maya merupakan teknologi yang dapat membuat penggunaanya dapat berinteraksi dengan apa yang ada di lingkup simulasi komputer [2]. Dari konsep VR ini, penelitian ini bertujuan untuk membuat sebuah peta interaktif tiga dimensi dengan objek Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember (ITS). Pembuatan VR dapat dilakukan dengan *Game Engine* seperti *Unreal Development Kit* (UDK) dan *Unity Engine*. Dalam penelitian ini akan menggunakan *Unity* sebagai *game engine* utama dalam pembuatan peta interaktif tiga dimensi.

Peta interaktif tiga dimensi sudah menjadi salah satu topik penelitian yang dilakukan oleh beberapa mahasiswa di ITS. Objek penelitian dalam penelitian ini, Jurusan Teknik Sipil ITS, sebelumnya sudah pernah dibuatkan peta interaktif tiga dimensi menggunakan UDK [3], namun setelah dilakukan wawancara dengan pihak Jurusan Teknik Sipil ITS, aplikasi tersebut dirasa memiliki kekurangan seperti mereka tidak tahu bagaimana bisa memanfaatkannya dan bagaimana dapat melakukan perubahan pada aplikasi tersebut. Penulis dari penelitian tersebut juga menyarankan beberapa hal untuk dikembangkan lagi, seperti bagaimana aplikasi tersebut dapat dijalankan secara *online*, karena aplikasi tersebut hanya dapat dijalankan di *desktop*. Penelitian lain menggunakan peta tiga dimensi yang menggunakan *Unity* sebelumnya [4] tidak pula mencantumkan bagaimana *Unity* dapat tersambung pada database dalam *hosting*. *Unity* sendiri sebenarnya memiliki fitur untuk dapat mengambil data dari sebuah halaman web dalam bentuk *string*, gambar, hingga video, dimana ketiga hal tersebut masih lebih mudah diatur dengan halaman PHP atau langsung melakukan perubahan pada halaman admin *database*. Dengan begitu isi konten dalam aplikasi *Unity* seperti *string*, gambar, dan video dapat lebih mudah diubah tanpa perlu merubahnya dan melakukan *build* lagi menggunakan *Unity Editor*.

Oleh karena itu, dapat diambil kesimpulan bahwa peta tiga dimensi tersebut perlu dibuat supaya dapat berbasis *web*, yang mana dapat digunakan secara umum, yakni dengan meletakkannya di sebuah *website*. *Game Engine* yang digunakan dalam penelitian ini, *Unity*, memiliki fitur untuk membuat aplikasi yang dapat dijalankan dalam *internet browser* (seperti *Internet Explorer*, *Mozilla Firefox*, *Google Chrome*, dan sebagainya) menggunakan WebGL, yang mana tidak disediakan dalam UDK (saat penelitian ini berlangsung), dan memanfaatkan fitur pada *game engine* untuk dapat mengambil data dari *web hosting database*.

1.2 Rumusan Masalah

Masalah – masalah yang perlu dipecahkan dalam tugas akhir ini adalah sebagai berikut ini:

1. Bagaimana membuat halaman admin dalam bentuk PHP untuk dapat mengatur beberapa isi konten dari peta interaktif 3D.
2. Bagaimana cara membuat interaksi berupa simulasi maupun video dari peta interaktif 3D Jurusan Teknik Sipil ITS pada 6 laboratorium di Jurusan Teknik Sipil ITS, mengecualikan 3 laboratorium pada D3 Teknik Sipil ITS.
3. Bagaimana peta interaktif 3D dapat dimasukkan dan dijalankan menggunakan *internet browser* dan dapat ditaruh pada *website*.

1.3 Batasan Masalah

Dalam pengerjaan tugas akhir ini akan ada beberapa batasan masalah yang perlu diperhatikan yakni sebagai berikut ini :

1. Tools utama yang digunakan untuk membangun aplikasi peta interaktif 3D ini adalah *Unity 3D*.
2. Studi kasus dilakukan pada Jurusan Teknik Sipil ITS, mengecualikan D3 Teknik Sipil ITS yang terletak di Manyar, Surabaya.
3. Peta interaktif 3D yang dibuat tidak akan menampilkan ruangan atau barang yang bersifat privasi bagi Jurusan Teknik Sipil ITS.
4. Aplikasi yang dikembangkan tidak memuat *Artificial Intelligence* (AI).
5. Data yang bisa dirubah atau dimasukkan melalui halaman pengaturan secara langsung hanya dapat berupa video, gambar, dan tulisan. Hal-hal lain seperti objek dan animasi perlu dibuatkan dulu *assetbundles*-nya menggunakan *Unity Editor*.
6. Aplikasi yang dikembangkan tidak dapat dijalankan dalam *mobile browser*.

7. Aplikasi yang dikembangkan tidak memungkinkan pengguna untuk bertemu dengan pengguna lain (*multiplayer*) di dalam aplikasi.
8. Pengguna umum aplikasi tidak dapat merubah konten yang ada di dalamnya.

1.4 Tujuan Tugas Akhir

Tujuan dari penelitian ini adalah:

1. Membangun aplikasi yang dapat menunjukkan peta secara 3D dan dapat melakukan interaksi dalam lingkup peta 3D tersebut.
2. Membangun aplikasi yang dapat menunjukkan simulasi beberapa kegiatan Lab yang ada di dalam Jurusan Teknik Sipil ITS.
3. Membangun halaman pengaturan isi konten dalam bentuk web sehingga informasi di dalam aplikasi seperti gambar dan tulisan dapat diperbaharui.

1.5 Manfaat Tugas Akhir

Manfaat – manfaat yang akan didapatkan dari penelitian ini adalah sebagai berikut:

1. Manfaat untuk institusi :
 - Adanya pengembangan dari penelitian yang menggunakan *Unity* sebagai *game engine*.
2. Manfaat untuk klien (Jurusan Teknik Sipil ITS) :
 - Mendapatkan peta interaktif 3D dari Jurusan beserta simulasi dari beberapa kegiatan laboratorium.
 - Mendapatkan peta interaktif yang dapat merubah sebagian isi kontennya menggunakan halaman PHP sehingga tidak harus melakukan build lagi menggunakan *Unity Editor*.
 - Mendapatkan peta interaktif 3D yang dapat ditaruh di website.

1.6 Relevansi

Topik pada tugas akhir ini memiliki keterkaitan atau relevansi dengan ketentuan topik tugas akhir pada Jurusan Sistem Informasi yakni sebagai berikut:

- Tugas akhir ini layak dijadikan sebagai tugas akhir pada tingkat S1, karena tugas akhir ini dapat memberikan solusi dalam pemecahan masalah terkait penghubungan aplikasi peta interaktif tiga dimensi dengan *web database* sehingga mempermudah adanya perubahan.
- Tugas akhir ini meliputi matakuliah pada Jurusan Sistem Informasi seperti Rancang Bangun Perangkat Lunak dan Interaksi Manusia dan Komputer. Dengan begitu penelitian tugas akhir ini masih dapat bersangkutan dengan Jurusan Sistem Informasi.

Halaman ini sengaja dikosongkan

BAB II

TINJAUAN PUSTAKA

Pada Bab ini akan dipaparkan tinjauan pustaka terkait dengan permasalahan yang dibahas di Tugas Akhir.

2.1 Studi Sebelumnya

Pada tugas akhir yang mencakup topik peta 3d, sebelumnya sudah pernah didapati penelitian yang dilakukan terkait pengembangan aplikasi peta tiga dimensi. Penelitian sebelumnya yang difokuskan disini adalah penelitian yang dilakukan pada Jurusan Teknik Sipil Its yang menggunakan UDK[3]. Pada penelitian tersebut didapatkan saran bahwa dibutuhkan bagaimana supaya aplikasi tersebut dapat digunakan secara online, ditambah lagi wawancara yang dilakukan kepada Jurusan Teknik Sipil ITS yang juga setuju dengan saran tersebut, karena mereka sebelumnya setelah dibuatkan peta interaktif 3D menggunakan UDK, mereka tidak tahu bagaimana cara menggunakan dan merubahnya. Pada saat penelitian ini dilakukan, *Unity* sudah memiliki fitur untuk mengambil data dari internet, yakni dengan menggunakan fitur dengan nama WWW. Dengan adanya fitur ini maka aplikasi yang dibuat menggunakan *Unity* dapat mengambil suatu teks (*string*), gambar, ataupun video melalui URL. *GameObject* dan animasi juga dapat diambil menggunakan fitur WWW dengan melakukan kompresi terlebih dahulu dengan fitur *assetBundles* yang ada di dalam *Unity Editor*. Fitur *AssetBundles* sudah ada semenjak *Unity 4*, hanya saja fitur tersebut hanya tersedia pada versi berbayar, dan para pengguna yang tidak berbayar baru bisa menggunakan fitur tersebut ketika *Unity5* diluncurkan pada Maret 2015.

Penelitian sebelumnya, baik yang menggunakan UDK[3] maupun *Unity*[4] tidak memanfaatkan *web database*, sehingga apabila ingin merubah suatu konten yang ada di dalam aplikasi yang sudah jadi, maka perlu membuka kembali *Unity Project* menggunakan *Unity Editor* dan kembali melakukan build

project. Dalam melakukan *build*, waktu yang digunakan tidak sebentar apabila sudah terdapat banyak konten di dalam proyeknya. Jika konten yang ingin dirubah hanya sebagian teks, gambar, ataupun video, hal ini dirasa tidak efisien. Melihat hal tersebut maka perlu dilakukan pengembangan untuk mengambil data-data yang berada pada *web database*. Dengan demikian jika ingin ada isi konten sederhana yang ingin digantikan, bisa cukup dengan membuka halaman web yang dapat mengatur isi dari database tersebut, dan lalu membuat halaman web lain yang dapat menampilkan konten tersebut yang nantinya akan diambil oleh *Unity*.

2.2 Dasar Teori

Pada bagian ini akan dijelaskan berbagai dasar teori yang digunakan untuk menjadi dasar dari penelitian ini. Dasar teori yang digunakan adalah sebagai berikut:

2.2.1 Peta Tiga Dimensi

Peta adalah salah satu bentuk dokumen yang memuat informasi tentang hubungan spesial unsur-unsur yang ada di muka bumi, yang menggambarkan dunia nyata di atas bidang datar dalam bentuk simbol-simbol dan skala tertentu yang dapat dipertanggungjawabkan secara matematis[5]. Tiga dimensi yang biasa disingkat 3D adalah bentuk yang memiliki panjang, lebar, dan tinggi. Istilah ini banyak digunakan dalam bidang seni, animasi, komputer, dan matematika [6]. Sehingga dapat disimpulkan bahwa objek tiga dimensi adalah objek dengan bentuk tertentu yang memiliki panjang, lebar, dan tinggi. Saat ini objek tiga dimensi dapat dibentuk dalam lingkungan virtual atau maya yang tidak dapat disentuh secara langsung.

Peta tiga dimensi mengadopsi ketentuan yang berlaku dari objek secara tiga dimensi. Dimana informasi spasial dan unsur-unsur yang ada ditampilkan dengan bentuk tiga dimensi baik pada lingkungan nyata maupun maya atau virtual. Dengan skala yang telah ditentukan sebelumnya, informasi yang ada pada peta tiga dimensi ditampilkan dengan pertimbangan kondisi dan keadaan nyata dari tempat dan lokasi tersebut.

2.2.2 *Game Engine*

Game Engine dipahami sebagai entitas yang tunggal dan tertutup merepresentasikan seluruh aspek yang dapat digeneralisasikan terkait permainan (*game*), dimana memiliki hubungan dengan dua entitas lain yaitu konten *game* dan *tools* untuk pengembangan *game* [7]. *Game Engine* merupakan sistem perangkat lunak yang didesain untuk pembuatan dan pengembangan *video games*. Diantara fungsi utama dan *game engine* adalah melakukan *graphic processing* dalam hal ini biasa disebut dengan gambaran dari informasi seperti tekstur, pencahayaan bayangan, *collision detection* (metode perhitungan fisika ketika terjadi benturan antara 2 objek), dan pengaturan suara [6].

Selain itu dengan adanya *game engine*, dapat memberikan kemudahan dalam melakukan pengembangan dan eksekusi sebuah *game*, dimana *game engine* berlaku sebagai penggerak dari sistem dan *game* yang dibuat tersebut.

2.2.3 *Unity Game Engine*

Unity adalah ekosistem pengembangan *game* serta sebagai *engine* untuk *rendering* terintegrasi dengan sebuah set lengkap alat yang intuitif dan alur kerja yang cepat untuk membuat 3D interaktif dan konten 2D. *Unity3D Engine* merupakan salah satu teknologi yang digunakan untuk membangun sebuah *game*. *Unity Engine* memiliki kerangka kerja (*framework*) lengkap untuk pengembangan profesional. Sistem inti *engine* ini menggunakan beberapa pilihan bahasa pemrograman, diantaranya C#, javascript, dan boo. Seperti kebanyakan *game engine* lainnya, *Unity Engine* dapat mengolah beberapa data seperti objek tiga dimensi, suara, tekstur, dan lain sebagainya. *Unity Engine* atau *Unity Game Engine* sebagai editor digunakan untuk pengembangan dan perancangan konten untuk kemudian dieksekusi sebagai produk akhir. Metode utama yang digunakan *Unity Engine* adalah *integrated graphical environmental*. *Unity Game Engine* mirip dengan *game engine*

lainnya seperti *Director*, *Blender Game Engine*, *Virtools*, *Torque*, *Game Builder*, atau *Gamestudio*.

Berikut fitur *Unity* yang dapat digunakan dalam tugas akhir ini :

1. Menggunakan *Unity* untuk membangun level (*drag-and-drop* objek yang telah dibuat sebelumnya dengan *Blender*).
2. Fitur programming. *Unity* memiliki fitur untuk menampilkan materi secara menakjubkan, namun *Unity* tidak memiliki banyak fungsi yang dapat digunakan untuk langsung melakukan interaksi, maka dari itu dibutuhkan *programming* dahulu untuk dapat menyediakan interaksi dalam simulasi *game*.
3. Fitur *audio reverb zone*. Fitur yang membuat suara dalam suatu lokasi atau area mengeluarkan suara yang berbeda, tergantung pada jarak yang telah ditentukan, sehingga pengguna akan mendengarkan suara yang berbeda-beda pada setiap jarak.
4. Fitur *skybox* yang dapat mengubah langit pada *game*.
5. Fitur *Assets* untuk menyimpan asset yang bisa diakses oleh *Unity Editor*.
6. Fitur *AssetBundles* untuk dapat menyimpan objek yang ditaruh pada *Unity* untuk bisa digunakan kemudian. *Bundle* yang dibuat dapat ditaruh di *web*.
7. Fitur WWW untuk dapat mengambil data dari *web*.
8. Fitur *User Interface* (UI) untuk dapat membentuk tampilan UI bagi pengguna.

2.2.4 Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember

Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember (ITS) merupakan salah satu jurusan yang dimiliki Fakultas Teknik Sipil dan Perencanaan (FTSP) ITS. Jurusan Teknik Sipil ITS mengusung visi sebagai pusat rujukan bidang Teknik Sipil di Indonesia yang menunjang pembangunan infrastruktur yang inovatif dan bereputasi internasional. Selain memperoleh akreditasi A oleh BAN-PT, Teknik Sipil ITS adalah Jurusan

pertama di Indonesia yang mendapatkan sertifikasi ISO 9001:2000 di bidang akademik sejak tahun 2007 hingga saat ini. Program studi yang ditawarkan oleh Jurusan Teknik Sipil ITS adalah program D3, D4, S1 Reguler, S1 Lintas Jalur, S2, dan S3. Teknik Sipil ITS saat ini didukung oleh 9 laboratorium antara lain Laboratorium Hidroteknik dan Pantai, Laboratorium Perhubungan dan Bahan Konstruksi Jalan, Laboratorium Beton dan Bahan Bangunan, Laboratorium Mekanika Tanah dan Batuan, Laboratorium Struktur, Laboratorium Manajemen Konstruksi, Laboratorium Uji Material, Laboratorium Model Teknik Sipil, dan *Workshop/Bengkel*.

Sebagian dari gedung A, B, dan C yang terletak di sekitaran Teknik Sipil ITS digunakan oleh pihak Jurusan Arsitektur ITS. Adapun fungsi dan gedung yang digunakan untuk masing-masing bangunan beserta tingkatan lantainya adalah sebagai berikut:

- Gedung A: Pada lantai dasar Gedung A terdapat Laboratorium Beton dan Bahan Bangunan dan lantai setelah ini digunakan oleh pihak Jurusan Arsitek ITS
- Gedung B: Pada lantai dasar gedung B, 2 ruangan digunakan sebagai Laboratorium Struktur dan Laboratorium Manajemen Konstruksi
- Gedung C: Pada lantai dasar gedung C, 2 ruangan digunakan sebagai Laboratorium Perhubungan dan Bahan Konstruksi Jalan dan Laboratorium Mekanika Tanah dan Batuan. Ruangan-ruangan yang berada pada tingkat selanjutnya digunakan oleh Jurusan Arsitektur ITS.
- Gedung D: Pada gedung D ini digunakan sebagai *Workshop* dimana mesin-mesin untuk Laboratorium Struktur diletakkan dan digunakan untuk melakukan praktikum dan uji coba.
- Gedung E: Pada lantai dasar Gedung E 2 ruangan digunakan sebagai ruang kelas dan 1 ruangan digunakan sebagai kelas *workshop*. Lantai 2 dari gedung E masih dilakukan renovasi dan direncanakan sebagai kelas untuk pasca sarjana Teknik Sipil ITS.

- Gedung H: Lantai dasar dari Gedung H digunakan sebagai Ruang Baca Jurusan Teknik Sipil ITS, dan juga sekretariat Jurusan Teknik Sipil ITS yang terletak pada lantai 1 dan 2 dari Gedung H.
- Gedung I: Lantai dasar dari Gedung I digunakan sebagai Kantin, Ruang Seni, Ruang Sekretariat Himpunan Mahasiswa, dan juga Ruang Kelas. Untuk lantai 2 digunakan sebagai ruang dosen.
- Gedung J: Lantai dasar dari Gedung J digunakan sebagai ruang kelas, dan lantai 2 dari Gedung J digunakan sebagai ruang dosen.
- Gedung L: Gedung L digunakan sebagai Laboratorium Hidroteknik dan Pantai.

2.2.5 Aplikasi *Modeling* 3D

Untuk membuat model tiga dimensi dari suatu objek, baik objek tersebut merupakan bangunan, alat, aktor, dan sebagainya, diperlukan aplikasi *modelling* 3D. Saat ini sudah terdapat beberapa aplikasi yang tersedia untuk diunduh dan digunakan, baik tersedia secara gratis (dengan atau tanpa batasan) ataupun berbayar. Beberapa dari aplikasi tersebut adalah *3DS Max*, *Blender*, *Sketch-Up*, *Maya*, dan sebagainya. Aplikasi *modelling* 3D yang akan digunakan dalam penelitian ini adalah *Blender*.

2.2.6 Program Pengolahan Gambar

Dalam penelitian ini diperlukan program yang dapat digunakan untuk mengolah objek gambar yang nantinya digunakan dalam membuat model 3D tertentu maupun untuk komponen 2D pada aplikasi. Berikut adalah contoh dari program pengolahan gambar yang akan digunakan dalam penelitian ini :

- Adobe Photoshop

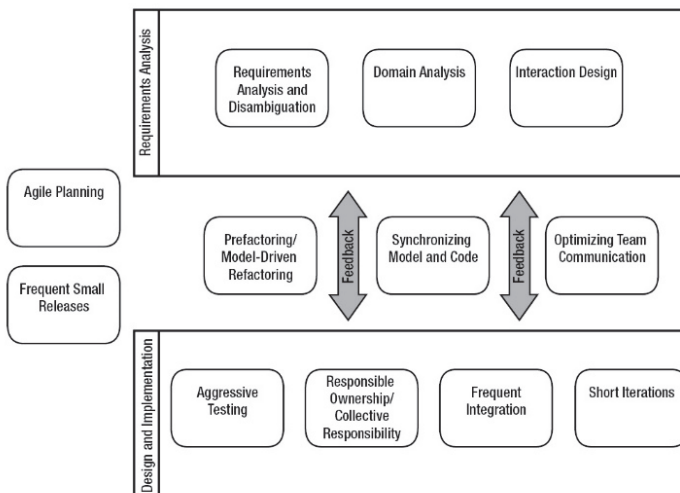
Adobe Photoshop adalah aplikasi dari Adobe Systems yang banyak digunakan dalam mengolah gambar. Adobe Photoshop memiliki banyak fungsi yang digunakan untuk mengolah gambar. Dalam pembuatan aplikasi ini, fungsi yang digunakan diantaranya adalah fungsi tekstur, lighting, dan filter

- Krita

Krita merupakan aplikasi pengolahan gambar *open-source* yang dapat diunduh secara gratis di Krita.org. Krita memiliki fitur-fitur yang hampir sama dengan Adobe Photoshop, dimana kita dapat mengolah gambar secara digital, dimana dapat digunakan baik untuk membuat ikon, tekstur, dan semacamnya. Dalam penelitian ini Krita digunakan sebagai cadangan karena *PC/Laptop* yang digunakan tidak semua memiliki Photoshop.

2.2.7 Agile ICONIX Practice

Agile ICONIX Practice mengumpamakan penggunaan *feedback* (umpan-balik) dalam praktiknya. Setiap tingkatan dari proses praktik ini dibalidasi oleh beberapa bentuk dari *feedback*, contohnya adalah review model seperti *robustness diagram* atau *unit-test* yang membantu untuk memvalidasi desain, maupun *feedback* dari klien maupun pengguna akhir dari prototipe awal [8]. Setiap proses dari *Agile ICONIX Practice* dapat digambarkan melalui gambar sebagai berikut:



Gambar 2-1 *Agile ICONIX Practice*

2.2.7.1 *ICONIX Process Practice*

Beberapa dari praktik-praktik berikut mencakup analisis untuk proses ICONIX dan desain, ditambah dengan melakukan desain untuk interaksi, dimana merupakan salah satu cara untuk mengendalikan perubahan dan mengurangi pergolakan dari kebutuhan yang ada [8].

a. *Requirements Analysis and Disambiguation*

Dalam tahap ini, pengembang (*developer*) berusaha untuk mencari dan menemukan segala kebutuhan (*requirements*) yang ada dengan berbicara kepada klien dan pengguna akhir. Inti dari tahap awal ini adalah dengan tujuan untuk menggunakan berbagai varian teknik, termasuk *robustness analysis* dan *domain analysis* untuk proyek, baik dari sisi pengembang maupun klien.

b. *Domain Analysis*

Proses *domain analysis* (analisa domain) dan *domain model* (model domain) dilakukan dalam tahap ini. Model domain digunakan untuk membantu secara tepat dalam menangkap dan menjelaskan kebutuhan proyek dengan disediakannya kosakata umum. Kosakata ini dapat dibagikan kepada seluruh orang yang terkait dalam proyek termasuk pengguna akhir.

c. *Interaction Design*

Tahap *Interaction design* atau desain interaksi dilakukan untuk membuat interaksi kepada penggunanya, dimana *User Interface* (antarmuka-pengguna) atau disingkat sebagai UI dibuat sehingga dapat mempermudah pengguna dalam melakukan interaksi. Tahap ini dilakukan pada bagian awal siklus hidup proyek. Hal tersebut dikarenakan desain UI dapat memicu efek mendalam pada pemahaman terhadap domain dari masalah. Penggunaan waktu pada tahap ini sebaiknya digunakan sebaik mungkin terhadap kebutuhan pengguna akhir, dan untuk menyelaraskan dengan kebutuhan bisnis dari klien. Dengan melakukan hal tersebut maka diharapkan untuk

dapat mengurangi kemungkinan pergolakan kebutuhan dan biaya untuk timbul.

d. *Prefactoring/Model-Driven Refactoring*

Prefactoring bertujuan untuk mempertimbangkan pola desain untuk menjadi yang terbaik terlebih dahulu sebelum melanjutkan untuk memulai proyek, sehingga meminimalisir kemungkinan untuk melakukan *refactoring* (mengulang kembali perancangan), sehingga bisa disebut juga sebagai *model-driven refactoring*, *refactoring* yang terpicu berdasarkan model-nya terlebih dahulu. Secara lebih detail, tahap ini adalah untuk melakukan pemetaan terhadap himpunan dari beberapa fungsi perangkat lunak yang telah diidentifikasi sebelumnya dengan tujuan untuk dapat mengimplementasikan kebutuhan-kebutuhan yang ada menjadi beberapa himpunan *class* yang telah ditemukan.

2.2.7.2 *Agile Practice*

Praktik berikut dapat dianggap sebagai *agile practice* tradisional [8]:

a. *Aggressive Testing*

Setiap penulisan kode dimulai, desain memiliki kemungkinan untuk setidaknya berubah sedikit. Dengan adanya hal tersebut, maka perlu dilakukan *refactoring*. Sebagai hasilnya nanti akan ditemukan sejumlah *unit test* yang sekiranya akan diperlukan untuk membantu penangkapan kesalahan pada saat melakukan perubahan.

b. *Synchronizing model and Code After Each Release*

Untuk setiap terbitan (*release*) produk, perlu dilakukan sinkronisasi antara *code* dengan desain yang ada untuk hal-hal yang dianggap perlu.

c. *Agile Planning*

Proyek yang bersifat *agile* adalah proyek yang terdorong oleh perencanaan yang dibuat saat ini daripada proyek yang

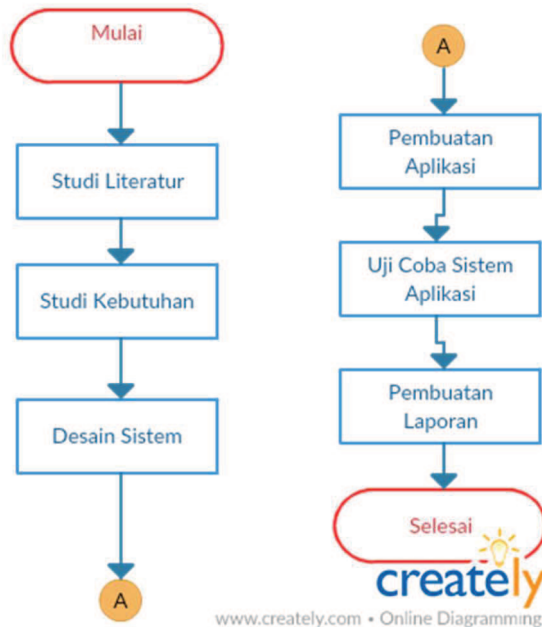
didorong oleh perencanaan yang telah dibuat sebelumnya. Proyek *agile* seperti ini akan memiliki lebih banyak perencanaan yang berlangsung dengan tahapan yang lebih sedikit dan detil yang lebih baik

BAB III METODOLOGI

Bab ini akan menjelaskan tentang tahapan – tahapan dalam pengerjaan tugas akhir sesuai dengan metodologi yang dipakai yakni, menggunakan metode *prototyping*.

3.1 Tahapan Pelaksanaan Tugas Akhir

Pada bagian ini akan dijelaskan tentang metodologi yang akan digunakan dalam penyusunan tugas akhir. Metodologi akan digunakan sebagai panduan dalam penyusunan tugas akhir agar terarah dan sistematis. Urutan dari pengerjaan tugas akhir dapat dilihat pada gambar Gambar 3-1



Gambar 3-1 *Flow Chart* tahap pengerjaan tugas akhir

3.1.1 Studi Literatur

Studi literatur yang dilakukan adalah pembelajaran dan pemahaman literatur yang berkaitan dengan permasalahan yang ada. Beberapa yang akan dipelajari seperti cara mengoperasikan *Unity Game Engine* dan memanfaatkannya untuk membuat peta, serta cara penggunaan perangkat lunak lainnya yang mendukung pengembangan aplikasi ini, dan juga untuk dapat membantu menentukan kebutuhan-kebutuhan yang masih kurang dari penelitian sebelumnya.

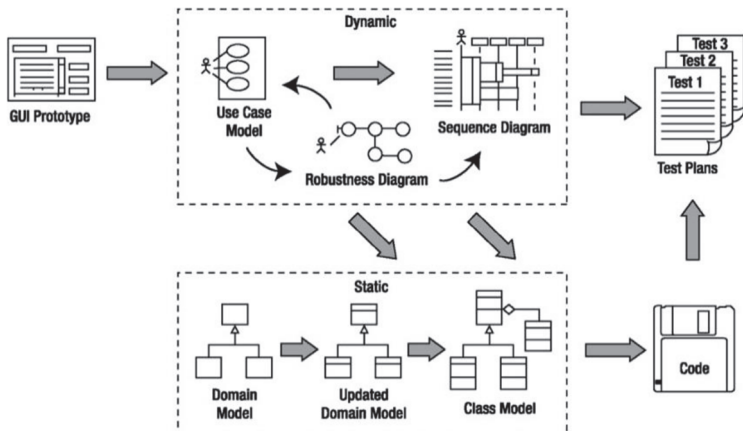
3.1.2 Studi Kebutuhan

Pada tahap ini akan dilakukan analisis kebutuhan dengan harapan untuk dapat menemukan beberapa kebutuhan yang akan dispesifikasikan sebagai kebutuhan fungsional dan non-fungsional. Dalam melakukan perancangan sistem, kebutuhan sistem perlu dispesifikasikan terlebih dahulu. Pada bagian ini akan dijelaskan mengenai kebutuhan-kebutuhan perangkat lunak, baik dari kebutuhan fungsional, kebutuhan non-fungsional, maupun aktor-aktor yang terlibat dalam sistem. Kebutuhan tersebut akan bisa didapatkan melalui wawancara dengan pihak Jurusan Teknik Sipil ITS mengenai kebutuhan mengenai aplikasi peta interaktif yang akan dibuat dan bagian daerah mana sajakah dari Jurusan Teknik Sipil ITS yang boleh dimasukkan dalam peta interaktif, kendala dari penelitian peta interaktif sebelumnya untuk Jurusan Teknik Sipil ITS yang menggunakan UDK, wawancara dengan masing-masing pihak dari laboratorium teknik sipil mengenai interaksi yang bisa dilakukan di dalam laboratorium pada Jurusan Teknik Sipil ITS, dan dengan melakukan studi mengenai penelitian sebelumnya. Untuk data tambahan lain adalah dengan meminjam denah gedung, dan mengambil foto dari gedung dan lokasi sekitar Jurusan Teknik Sipil ITS untuk membantu dalam pembuatan model 3D dan interaksi dalam peta interaktif tiga dimensi.

Setelah didapatkan data dan informasi yang mencukupi, maka akan dilakukan penyusunan konsep kebutuhan fungsional sistem, kebutuhan non-fungsional sistem, analisis pemilihan tombol dan navigasi, dan daftar aktor dalam sistem.

3.1.3 Desain Sistem

Pada tahap ini akan dilakukan perancangan aplikasi untuk peta interaktif tiga dimensi dari Jurusan Teknik Sipil ITS. Proses permodelan sistem informasi dilakukan dengan menggunakan metode *ICONIX Process*, dimana proses tersebut diadopsi berdasarkan metode *Agile ICONIX Practice*. Gambaran dari langkah-langkah *ICONIX Process* divisualisasikan pada gambar 3-2.



Gambar 3-2 *ICONIX Process*

3.1.3.1 Pembuatan *GUI Storyboard*

Dalam tahap pembuatan *GUI Storyboard* akan dibuat prototipe yang mempresentasikan antarmuka pengguna dari aplikasi yang akan dibangun, sehingga dapat membantu penjabaran model dan *case* yang diperkirakan akan ada di dalam sistem. Prototipe akan dibuat sederhana, mengikuti tampilan antarmuka pengguna yang digunakan baik dari penelitian sebelumnya ataupun *video game* secara umum.

3.1.3.2 Pembuatan *Domain Model*

Domain model terdiri dari sejumlah entitas atau objek yang ada di dalam aplikasi beserta relasi yang menghubungkan antar entitas atau objek tersebut. *Domain model* dibuat berdasarkan relasi objek utama dengan beberapa komponen lain dan akan terus diperbarui sesuai dengan pembangunan sistem.

3.1.3.3 Pembuatan *Use Case*

Use Case akan berperan untuk menjelaskan interaksi yang terjadi antara aktor dengan sistem yang ada. Pembuatan *Use Case* akan dilakukan berdasarkan kebutuhan fungsional sistem yang telah didapatkan dari studi kebutuhan sistem.

3.1.3.4 Pembuatan *Robustness Analysis*

Setelah didapatkan *Use Case* maka akan *Robustness Analysis* menganalisa tahap-tahap dari *use case* apakah sudah tepat, sehingga *use case* yang telah dibuat tidak ambigu.

3.1.3.5 Pembuatan *Sequence Diagram*

Dalam tahap ini akan dilakukan penghubungan antara *domain model* yang masih berisi data dan entitas namun masih belum ada penjelasan bagaimana entitas yang ada dapat saling bertukar data sesuai dengan skenario dari *use case*.

3.1.3.6 Pembuatan *Test Case*

Setelah didapatkan semua datanya maka akan dibuatkan beberapa skenario *test case* kebutuhan fungsional dan non-fungsional, sehingga akan didapatkan apakah kebutuhan fungsional dan non-fungsional dalam penelitian ini sudah terpenuhi.

3.1.4 Pembuatan Aplikasi

Dalam tahap ini, setelah didapatkan desain dari sistem, akan terlebih dahulu dibuatkan model 3D dari gedung Jurusan Teknik Sipil ITS dengan melihat referensi yang didapatkan

pada tahap studi kebutuhan seperti denah, foto gedung, dan bagian mana saja yang diperbolehkan untuk dibuatkan model tiga dimensinya, dan alat-alat apa saja yang sekiranya bisa dibuatkan animasi-nya untuk dimasukkan ke dalam peta interaktif. Dalam pembuatan model tiga dimensi dan animasinya akan digunakan aplikasi *Blender*. Setelah model gedung beserta isinya telah dibuat menggunakan *Blender*, maka model dan animasi tersebut sudah dapat dimasukkan ke dalam *folder asset* yang terdapat dalam folder proyek *Unity* sehingga dapat dilakukan *drag-and-drop* dalam *Unity Editor*. Untuk beberapa bagian yang bersifat terhubung pada *web database*, perlu dibuatkan dulu halaman PHP untuk mengambil dan menampilkan data yang bisa diambil oleh *Unity*, dan setelah data tersebut sudah bisa dimunculkan, maka sudah dapat dibuatkan *script* dalam *Unity* yang berperan untuk melakukan perubahan sebagian konten dalam *Unity* berdasarkan data yang telah diambil dari *database*.

3.1.5 Uji Coba Sistem Aplikasi

Setelah dilakukan pembuatan aplikasi dan dirasa sudah mencukupi dari rancangan awal, akan dilakukan uji coba untuk mengetahui apakah aplikasi tersebut sudah memenuhi kebutuhan-kebutuhan fungsional dan non-fungsional dengan menggunakan *test case* yang sudah dirancang pada tahap desain sistem.

3.1.6 Penulisan Laporan

Setelah tahapan pelaksanaan tugas akhir sudah dilakukan seluruhnya, maka akan dibuatkan laporan mengenai jalannya penelitian ini menyesuaikan dengan tahapan pelaksanaan tugas akhir.

Halaman ini sengaja dikosongkan

BAB IV PERANCANGAN

Dalam bab ini akan dijelaskan mengenai perancangan dari aplikasi peta interaktif tiga dimensi. Pada awalnya akan dilakukan analisis kebutuhan sistem terlebih dahulu dan lalu akan dilanjutkan dengan Proses permodelan sistem informasi menggunakan metode *ICONIX Process*.

4.1 Analisis Kebutuhan Sistem

Untuk melaksanakan desain aplikasi, perlu dilakukan analisis terlebih dahulu sehingga dapat ditemukan beberapa kebutuhan yang akan dispesifikasikan sebagai kebutuhan fungsional dan non-fungsional. Pada bagian ini akan dijelaskan mengenai kebutuhan-kebutuhan perangkat lunak, baik dari kebutuhan fungsional, kebutuhan non-fungsional, maupun aktor-aktor yang terlibat dalam sistem. Kebutuhan tersebut bisa didapatkan melalui wawancara dengan pihak Jurusan Teknik Sipil ITS dan melihat penelitian sebelumnya. Setelah cukup mendapatkan informasi, kemudian dilakukan penyusunan konsep kebutuhan fungsional dan non-fungsional. Namun demikian kebutuhan-kebutuhan tersebut dapat sedikit berubah sewaktu-waktu dalam pengujian aplikasi.

4.1.1 Kebutuhan Fungsional Sistem

Berdasarkan hasil pertemuan dengan salah satu aktor yang pernah menggunakan aplikasi Peta Interaktif Tiga Dimensi Jurusan Teknik Sipil menggunakan *Unreal Engine*, serta melihat hasil penelitian tersebut, maka didapatkan kebutuhan fungsional sebagaimana terlampir dalam tabel 4-1.

4.1.2 Kebutuhan Non-Fungsional Sistem

Selain dari kebutuhan fungsional dari aplikasi, berikut adalah kebutuhan non-fungsional mengenai aplikasi yang ditunjukkan pada tabel 4-2 untuk melengkapi kebutuhan yang meliputi performa dari aplikasi ini.

4.1.3 Analisis Pemilihan Tombol & Navigasi

Pengguna perlu menggunakan tombol untuk dapat melakukan navigasi dalam aplikasi Peta Interaktif 3D ini, maka dari itu dibuatlah analisis pemilihan tombol navigasi yang disesuaikan dengan kondisi umum permainan tiga dimensi. Analisis pemilihan tombol navigasi dicantumkan pada tabel 4-3

Tabel 4-1 Daftar Kebutuhan Fungsional Sistem

ID	Kebutuhan Fungsional
KF-01	Terdapat tampilan antarmuka pengguna umum untuk menu utama
KF-02	Terdapat profil Jurusan Teknik Sipil ITS
KF-03	Pengguna umum dapat menjelajahi peta 3D Jurusan Teknik Sipil ITS
KF-04	Pengguna umum dapat melihat informasi ruangan
KF-05	Pengguna umum dapat menjalankan simulasi mesin pengujian kuat Tarik beton
KF-06	Pengguna umum dapat menjalankan simulasi mesin pengujian kuat tekan beton
KF-07	Pengguna umum dapat menjalankan simulasi mesin pengujian kuat tekan besi
KF-08	Pengguna umum dapat melihat video informasi laboratorium
KF-09	Pengguna umum dapat menghentikan penjelajahan sementara (<i>pause</i>)
KF-10	Pengguna umum dapat melihat peta Jurusan Teknik Sipil ITS
KF-11	Administrator dapat memperbarui teks, gambar, atau video yang terdapat dalam aplikasi melalui <i>web</i> .

Tabel 4-2 Daftar Kebutuhan Non-Fungsional Sistem

ID	Kebutuhan Non-Fungsional
KNF-01	Kemampuan aplikasi untuk menampilkan rerata <i>frame</i> per detik (uji performa)
KNF-02	Kemampuan akses aplikasi serta lama <i>loading</i> yang dibutuhkan (pengujian <i>platform web</i>).
KNF-03	Kemampuan aplikasi untuk dapat diakses oleh beragam <i>web browser</i> (kompatibilitas <i>browser</i>).
KNF-04	Kemampuan akses aplikasi saat koneksi dengan server terputus
KNF-05	Kemampuan aplikasi saat diakses bersamaan oleh lebih dari satu pengguna
KNF-06	Kemampuan akses aplikasi pada sistem operasi berbeda
KNF-07	Kemampuan aplikasi ketika ada perubahan <i>database</i> saat pertengahan simulasi (ada video yang berubah, ada video baru)

Tabel 4-3 Analisis Pemilihan Tombol Navigasi

No.	Perintah	Tombol	Hasil	Alasan
1.	Bergerak ke depan	W atau panah atas	Menggerakkan karakter ke arah depan arah pandang kamera karakter	Umum digunakan dalam permainan 3D populer
2.	Bergerak ke kiri	A atau panah kiri	Menggerakkan karakter ke arah kiri arah pandang kamera karakter	Umum digunakan dalam permainan 3D populer
3.	Bergerak mundur	S atau panah bawah	Menggerakkan karakter ke arah belakang arah pandang kamera karakter	Umum digunakan dalam permainan 3D populer

No.	Perintah	Tombol	Hasil	Alasan
4.	Bergerak ke kanan	D atau panah kanan	Menggerakkan karakter ke arah kanan arah pandang kamera karakter	Umum digunakan dalam permainan 3D populer
5.	Melihat sekitar	Gerakan <i>mouse</i>	Mengubah arah pandangan untuk melihat lingkungan virtual sekitar sesuai dengan gerakan <i>mouse</i>	Umum digunakan dalam permainan 3D populer
6.	Melompat	Spasi	Memerintah karakter untuk melompat	Umum digunakan dalam permainan 3D populer
7.	Melakukan interaksi	E	Memunculkan interaksi pengguna, jalannya simulasi, dan sebagainya	Umum digunakan dalam permainan 3D populer
8.	Berlari (<i>sprint</i>)	Shift	Selama tombol ditahan, karakter dapat bergerak lebih cepat	Umum digunakan dalam permainan 3D populer

4.1.4 Daftar Aktor

Dalam aplikasi ini akan melibatkan 2 aktor, yakni administrator dan pengguna umum. Penjelasan dari kedua aktor tersebut akan dicantumkan pada tabel 4-4.

Tabel 4-4 Daftar Aktor

Aktor	Deskripsi
Administrator	Orang yang memiliki wewenang untuk mengelola aplikasi secara umum seperti mengelola teks, gambar, video, dan objek-objek lain di dalam aplikasi
Pengguna Umum	Orang-orang yang dapat menggunakan aplikasi peta 3D namun tidak dapat melakukan perubahan pada aplikasi tersebut.

4.2 Desain Sistem

Setelah melakukan analisis terhadap kebutuhan aplikasi Peta 3D Jurusan Teknik Sipil ITS, tahap selanjutnya adalah melakukan pembuatan desain sistem yang menyesuaikan dengan *ICONIX Process*.

4.2.1 Pembuatan *GUI Storyboard*

Prototipe *GUI Storyboard* yang akan dibuat dalam penelitian ini akan terfokus pada menu-menu utama yang akan muncul di dalam aplikasi.

4.2.1.1 Prototipe Menu Utama

Antarmuka yang akan dimunculkan pertama kali adalah menu utama saat membuka aplikasi yang dikenal juga sebagai *main menu*, dimana antarmuka pada bagian ini akan berfungsi untuk bagian awal pengguna temui sebelum bernavigasi ke dalam peta. Dalam antarmuka ini diberikan 2 buah *button*, dimana 1 *button* akan berperan untuk memulai navigasi dalam peta interaktif, dan 1 buah *button* yang berperan untuk menunjukkan informasi mengenai jurusan atau aplikasi peta tiga dimensi itu sendiri. Dalam menu *about* direncanakan untuk diberikan panel dengan judul dan teks dibawahnya dimana teks tersebut akan diisikan dengan penejelasan dari aplikasi atau mengenai

jurusan. Prototipe dari antarmuka menu utama dan menu *about* akan ditunjukkan pada gambar 4-1 dan 4-2.



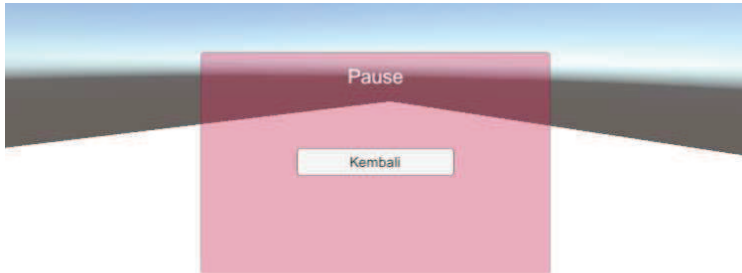
Gambar 4-1 Prototipe Menu Utama



Gambar 4-2 Prototipe Menu *About*

4.2.1.2 Prototipe *Pause Menu*

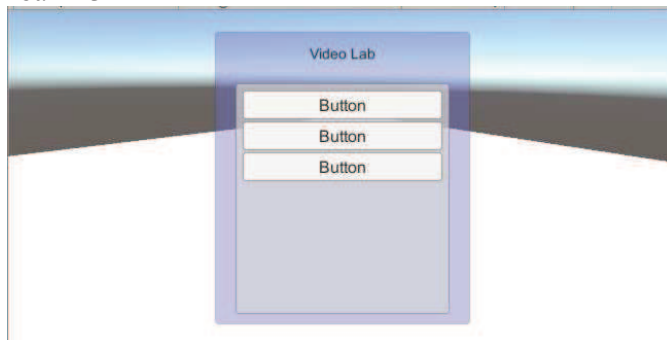
Antarmuka *pause menu* akan dimunculkan ketika suatu tombol ditekan untuk menghentikan permainan sementara dimana pengguna akan dapat memilih untuk kembali melanjutkan navigasi peta tiga dimensi, memilih penjelasan, atau keluar kembali kepada menu utama. Prototipe dari pause menu akan ditunjukkan pada gambar 4-3



Gambar 4-3 Prototipe *pause menu*

4.2.1.3 Prototipe Menu Daftar Video

Dalam menu daftar video ini akan menunjukkan daftar video yang ada mengenai suatu ruangan atau laboratorium. Pada bagian ini, akan terdapat beberapa tombol untuk masing-masing video, dan jumlah dari tombol tersebut akan berubah tergantung dari *web database* mengenai data video yang ada di dalam lab tersebut. Desain dari daftar video tersebut ditunjukkan pada gambar 4-3

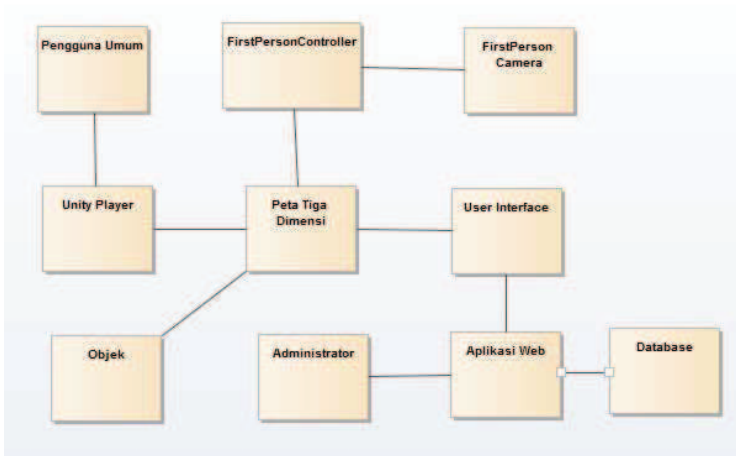


Gambar 4-4 Prototipe daftar video laboratorium

4.2.2 Pembuatan *Domain Model*

Domain model terdiri dari sejumlah entitas atau objek yang ada di dalam aplikasi beserta relasi yang menghubungkan antar

entitas atau objek tersebut. *Domain model* akan terus diperbarui sesuai dengan pembangunan sistem. *Domain Model* mengenai penelitian ini ditunjukkan pada gambar 4-5.



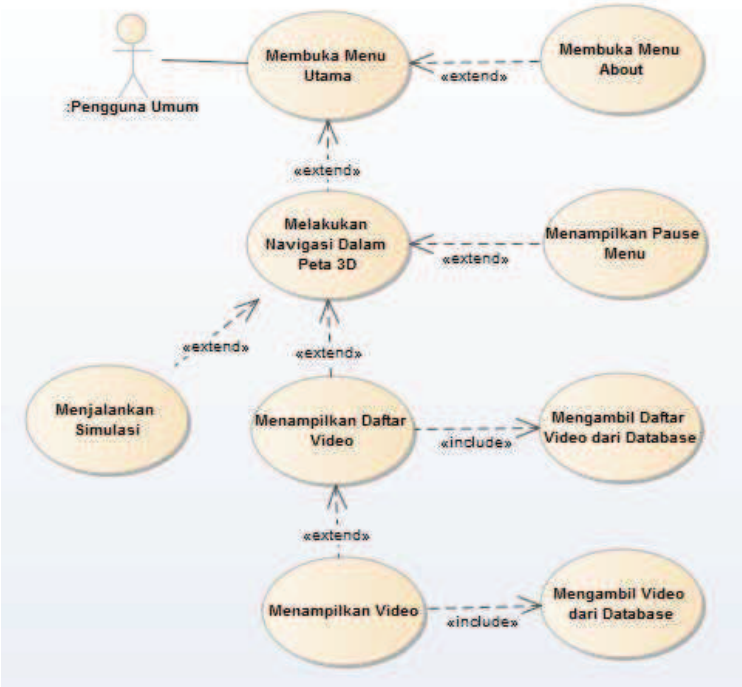
Gambar 4-5 Domain Model

4.2.3 Pembuatan Use Case

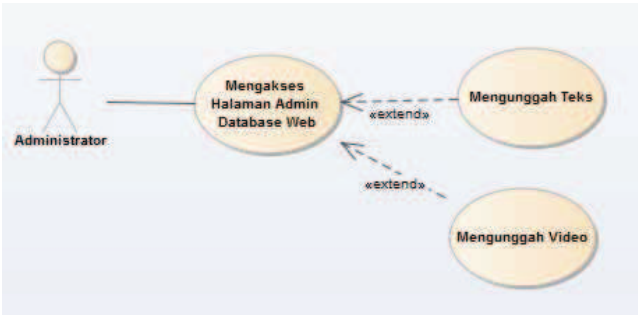
Untuk pembuatan rancangan *use case* harus sesuai dengan kebutuhan fungsional yang telah ditentukan sebelumnya, dan sedari itu dapat dibuat beberapa *use case*. Pemetaan *use case* dengan deskripsinya untuk aplikasi peta 3D ini beserta kebutuhan fungsional yang menjadi dasarnya dapat dilihat pada tabel 4-5, dan untuk penggambaran *use case model* bagi pengguna umum dan administrator, secara urut dapat dilihat pada gambar 4-6 dan 4-7, dan untuk *use case description* bagi masing-masing *use case* akan dilampirkan pada lampiran A.

Tabel 4-5 Pemetaan *Use Case* berdasarkan kebutuhan fungsional

No.	ID <i>Use Case</i>	Nama <i>Use Case</i>	ID Fungsional
1.	UC-101	Pengguna umum membuka menu utama	KF-01
2	UC-102	Pengguna umum membuka menu <i>about</i>	KF-02
3	UC-103	Pengguna umum melakukan navigasi dalam peta 3D	KF-03
4	UC-104	Pengguna umum menjalankan simulasi mesin	KF-05, KF-06, KF-07
5	UC-105	Pengguna umum menampilkan daftar video dari laboratorium	KF-04, KF-08
6	UC-106	Pengguna umum mendapatkan daftar video berdasarkan database	KF-08, KF-11
7	UC-107	Pengguna umum menampilkan video	KF-08
8	UC-108	Pengguna umum mendapatkan video dari database	KF-08, KF-11
9	UC-109	Pengguna umum menampilkan Pause Menu	KF-09
10	UC-110	Pengguna umum menampilkan <i>mini map</i>	KF-10
11	UC-201	Administrator Mengakses Halaman Pengaturan <i>Web Database</i>	KF-11
12	UC-202	Administrator mengunggah teks ke <i>database</i>	KF-11
13	UC-203	Administrator mengunggah video ke <i>database</i>	KF-11



Gambar 4-6 Use Case Model untuk Pengguna Umum



Gambar 4-7 Use Case Model untuk Administrator

4.2.4 Pembuatan *Robustness Analysis*

Stereotipe yang akan digunakan dalam *robustness analysis* penelitian ini antara lain adalah *boundary*, *control*, dan *entity*. Untuk gambaran *robustness diagram* dari *use case* akan ditunjukkan pada lampiran B.

4.2.5 Penggunaan *Sequence Diagram*

Sequence Diagram berfungsi untuk menjelaskan alur kerja aplikasi pada tiap *use case* dengan lebih mendetail, disertai dengan proses yang terjadi antar masing-masing elemen dalam diagram berdasarkan alur yang telah dijabarkan. Alur yang dijabarkan meliputi *basic path* dan *alternate path*. *Sequence diagram* untuk masing-masing *use case* dapat dilihat pada lampiran C.

4.2.6 Penggunaan *Test Case*

Di dalam *test case* dilakukan pengujian yang sesuai dengan desain yang telah dibuat sebelumnya. Tujuan dari perancangan *test case* adalah untuk menjaga sistem supaya tetap sesuai dengan desain yang telah dirancang. Daftar dari *test case* akan dilampirkan pada lampiran D.

Halaman ini sengaja dikosongkan

BAB V

IMPLEMENTASI DAN UJICOB

Di dalam bab ini akan dibahas hal-hal mengenai rancangan dan implementasi sistem yang dibuat dalam aplikasi simulasi peta 3D ini. Desain aplikasi ini dibuat berdasarkan kebutuhan fungsionalitas dan non-fungsionalitas. Kebutuhan fungsionalitas yang akan dibahas didefinisikan sebagai berikut:

- Melihat dan melakukan navigasi pada peta tiga dimensi
- Berinteraksi dengan objek
- Interaksi mengenai informasi khusus yang dimiliki oleh setiap lokasi
- Melihat video yang terdapat dalam *database web* dengan menggunakan *Unity*

Untuk kebutuhan non-fungsionalitas, akan didefinisikan sebagai berikut:

- *Hardware* (perangkat keras)
- *Software* (perangkat lunak) utama, yakni *Unity5*
- Aplikasi pendukung lainnya yang dibutuhkan

5.1 Lingkungan Implementasi

Berikut adalah spesifikasi komputer yang digunakan dalam pengembangan dan implementasi aplikasi peta 3D ini dapat dilihat pada Tabel

Tabel 5-1 Spesifikasi Komputer I

SPESIFIKASI KOMPUTER I	
CPU	Intel® Core™ i3-3110M CPU @ 2.40GHz (4 CPUs), ~2.4GHz
RAM	4096MB RAM
GPU	Intel® HD Graphics 4000; Memory 1792MB
Sistem Operasi	Windows 8.1 Single Language 64-bit (6.3, Build 9600)

Tabel 5-2 Spesifikasi Komputer II

SPESIFIKASI KOMPUTER II	
CPU	Intel® Core™ 2 Duo CPU E7500 @ 2.93GHz (2 CPUs), ~2.9GHz
RAM	4096MB RAM
GPU	AMD Radeon HD 6570 2805MB
Sistem Operasi	Windows 7 Professional 64-bit (6.1, Build 7601)

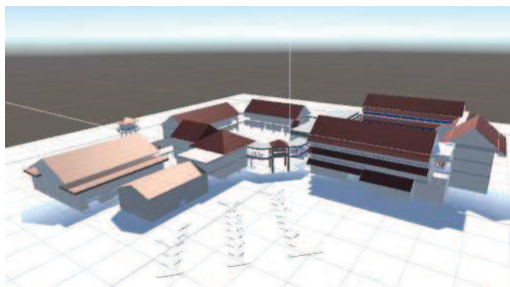
Perangkat lunak yang digunakan sebagai perangkat lunak utama adalah *Unity3D* untuk membuat aplikasi. Untuk perangkat lunak pendukung lainnya yang digunakan adalah *Blender* untuk membuat model dan beberapa animasi 3D dari gedung dan sebagian mesin yang terdapat di Jurusan Teknik Sipil ITS dan *Adobe Photoshop* sebagai aplikasi pendukung untuk mengolah gambar.

5.2 Implementasi Aplikasi

Pada bagian ini akan dijelaskan bagaimana pengerjaan dari pembuatan aplikasi akan dilaksanakan dari awal hingga akhir.

5.2.1 Model 3D Gedung Jurusan Teknik Sipil ITS

Gambar berikut menunjukkan hasil jadi model 3D dari gedung Jurusan Teknik Sipil ITS yang dibuat menggunakan aplikasi *Blender* dan telah ditaruh dalam *Unity Editor*.

**Gambar 5-1 Model 3D Gedung Jurusan Teknik Sipil ITS**

5.2.1.1 Perancangan dan Hambatan

Gedung Jurusan Teknik Sipil ITS merupakan salah satu gedung yang paling awal berdiri di lingkup kampus ITS, dimana gedung tersebut dibangun sejak tahun 1957. Dikarenakan hal tersebut, denah bangunan yang tersedia adalah dalam bentuk kertas dan tidak ada dalam bentuk mentahan file-nya ataupun hasil *scan* dari denah tersebut. Dikarenakan denah yang ada hanya dalam bentuk cetak, sehingga perlu dilakukan *scan* terlebih dahulu sehingga denah tersebut lebih mudah dibaca dan juga bisa dimasukkan pada *Blender* yang kemudian dapat membantu untuk membentuk model 3D. Dikarenakan denah tersebut adalah denah awal yang digunakan pada saat awal Gedung Teknik Sipil ITS didirikan, maka besar kemungkinan ada perbedaan dari denah dan lingkungan gedung saat ini, sehingga perlu dilakukan pemeriksaan kembali antara keduanya. Denah yang dipinjamkan tersebut juga tidak mencakup beberapa gedung secara keseluruhan, karena beberapa bagian gedung tersebut dimiliki oleh Jurusan Arsitek ITS, sehingga ada bagian gedung yang untuk pembuatannya menggunakan perkiraan berdasarkan foto. Adapun sebenarnya dari pihak teknik sipil memiliki model 3D dalam bentuk *AutoCAD*, hanya saja pada bagian sekretariat tidak tahu dimana file tersebut disimpan, sehingga dalam penelitian ini, seluruh model bangunan harus dibuat dari awal menggunakan *Blender*.

5.2.1.2 Solusi

Model 3D Gedung Teknik Sipil ITS dibangun berdasarkan denah dan foto gedung terbaru menggunakan *Blender*. Dalam pembuatannya, denah akan digunakan untuk membantu memperkirakan panjang, tinggi, dan lebar gedung dan foto gedung terbaru dipakai untuk memastikan apakah ada bagian denah yang berbeda dengan kondisi saat itu. Setelah model 3D selesai dibuat menggunakan *Blender*, mode 3D tersebut dapat diekspor ke aplikasi *Unity* dengan 2 cara:

1. Model 3D yang dibuat menggunakan *Blender* tersimpan dengan ekstensi *.blend*. File *.blend* dapat langsung

ditaruh pada folder *Asset* unity tanpa perlu melakukan *export* sebagai file lain. Hanya saja dengan cara ini maka seluruh objek yang ada di dalam blender akan ikut *ter-export*, sehingga perlu dipastikan tidak ada objek yang tidak terpakai sehingga file model 3D dapat langsung digunakan. Animasi yang terunggah juga adalah animasi secara keseluruhan, sehingga animasi masih perlu dipotong terlebih dahulu pada *Unity Editor* apabila dibutuhkan.

2. *Export* model 3D di dalam Blender sebagai .fbx, dan jika menggunakan metode ini, *export* bisa dilakukan untuk beberapa objek yang dipilih pada blender saja. Animasi yang dibuat juga bisa ikut secara terpisah jika satuan animasinya dipotong dengan fitur *NLA Editor* yang terdapat di dalam *Blender*.

Setelah model diletakkan pada folder *asset* Unity, selanjutnya model-model tersebut sudah dapat dimunculkan pada Unity baik menggunakan *drag and drop* melalui *Unity editor* ataupun dipanggil melalui *script*.

5.2.2 Peta Dua Dimensi (2D)

Peta dua dimensi yang akan digunakan dalam tugas akhir ini diberikan dalam bentuk cetak oleh pihak Jurusan Teknik Sipil ITS dan dilakukan scan sehingga didapatkan dalam bentuk *softcopy*. Untuk hasil scan dari denah lengkap mengenai tersebut akan dilampirkan pada lampiran E. Denah-denah ini akan membantu untuk mendapatkan ukuran dari gedung seperti tinggi, panjang, dan lebar gedung. Denah sederhana mengenai tata letak gedung Jurusan Teknik Sipil, seperti yang ditunjukkan pada Gambar 5-2, juga diberikan untuk membantu menentukan lokasi gedung saat dibuatkan model tiga dimensinya. Untuk membantu mencari tahu detil dari bagian atap gedung dan juga untuk membandingkan tata letak yang diberikan dengan kondisi aslinya maka dibandingkan dengan menggunakan foto *drone* tampak atas gedung Jurusan Teknik Sipil seperti yang ditunjukkan pada Gambar 5-3.



5.2.3 Pembuatan Database

Dalam pengembangan peta interaktif 3D ini akan dibutuhkan database untuk menampung data-data seperti teks, gambar, dan video, yang mana data tersebut perlu dipersiapkan dahulu untuk dapat diambil melalui *Unity* menggunakan fitur WWW dalam *Unity*. Beberapa *database* yang diperlukan adalah untuk Laboratorium Sipil dan Video Laboratorium, dan lalu akan dibuatkan *view* yang menggabungkan kedua tabel *database* tersebut supaya lebih mudah dibaca dan untuk nantinya saat memasukkan *sql query* dalam *script* PHP tidak terlalu panjang. Untuk database laboratorium sipil hanya memerlukan data sederhana seperti ID, nama laboratorium, dan keterangan dari laboratorium tersebut jika suatu saat diperlukan.

Setelah didapatkan tabel laboratorium sipil beserta isi-nya, maka dibuatkan tabel *database* untuk video laboratorium dengan ID laboratorium sebagai *foreign key* untuk tabel ini. Isi dari tabel video laboratorium ini adalah ID Video yang merupakan *Primary Key* untuk mempermudah menandakan video yang berbeda, Judul dari video tersebut, ID dari laboratorium yang berhubungan dengan video tersebut (*foreign key*), tipe dari video yang disimpan, dan video tersebut dalam bentuk enkripsi *btye64*. Untuk membantu visualisasi antara hubungan tabel tersebut dan struktur dari masing-masing table akan ditunjukkan pada Gambar 5-4.



Gambar 5-4 Visualisasi Foreign Key antara video_lab dengan laboratorium_sipil

Selanjutnya setelah didapatkan kedua tabel tersebut, maka perlu dibuatkan *view* di dalam *database* mengenai gabungan dari kedua tabel tersebut, sehingga ketika dalam pembuatan

halaman PHP nanti, jika membutuhkan data dari kedua tabel tersebut, tidak perlu memasukkan *sqlquery* yang panjang. Untuk menggabungkannya, akan dijalankan dahulu *sqlquery* sebagaimana ditunjukkan pada Potongan Kode 5-1.

```
CREATE VIEW `fullview_video_lab` AS
SELECT id_video
      ,judul_video
      ,lab.id_lab
      ,nama_lab
      ,Vid_Type
      ,Vid_64
FROM `laboratorium_sipil` AS lab
INNER JOIN `video_lab` AS vidlab
ON lab.id_lab = vidlab.id_lab;
```

Potongan Kode 5-1 *Query* untuk membuat ``fullview_video_lab``

Adapun *view* perlu dibuat satu lagi, yakni sama dengan ``fullview_video_lab``, hanya saja tanpa mengambil tipe video ataupun video dalam bentuk *byte64*. Hal ini ditujukan untuk mempercepat pengambilan daftar video dalam aplikasi *Unity*, karena tipe video dan video dalam bentuk *byte64* tidak dibutuhkan ketika hanya mengambil daftar nama video. Untuk *query* pembuatan *view* yang dimaksud dengan nama ``no64_videoLab``, dapat dilihat pada Potongan Kode 5-2.

```
CREATE VIEW `no64_videoLab` AS
SELECT id_video
      ,judul_video
      ,lab.id_lab
      ,nama_lab
FROM `laboratorium_sipil` AS lab
INNER JOIN `video_lab` AS vidlab
ON lab.id_lab = vidlab.id_lab;
```

Potongan Kode 5-2 *Query* untuk membuat ``no64_videoLab``

5.2.4 Pembuatan Halaman PHP

Halaman PHP dalam penelitian ini akan berperan untuk mengunggah data dan sebagai penghubung antara aplikasi *Unity* dengan *database* yang ada.

5.2.4.1 Pembuatan Halaman Pengunggah

Untuk dapat menyimpan video atau gambar di dalam database, maka data tersebut akan perlu dilakukan enkripsi dulu ke dalam *byte64*, sehingga mendapat bentuk teks yang bisa disimpan di dalam *database*. Untuk *source code* PHP yang berperan untuk mengunggah dan memperbarui yang dinamakan "*upload_videoLab.php*". Pada baris awal akan diperlukan kode untuk tersambung ke dalam *database* terlebih dahulu seperti pada Potongan Kode 5-3. Pastikan untuk memberikan *ini_set()* di awal untuk mengatur *memory_limit* untuk mencegah *error* yang menyatakan *memory exhausted* atau semacamnya.

```
<?php
ini_set('memory_limit', '1024M');
// Upload Video ke video_Lab
$servername = "localhost";
$username = "root";
$password = "";
$dbname = "db_unity";

// Create Connection //
$conn = new mysqli($servername, $username, $password, $dbname);

// cek koneksi //
if($conn->connect_error)
{
    die("connection failed : " . $conn->connect_error);
}
```

Potongan Kode 5-3 *upload_videoLab.php* melakukan koneksi ke database

Setelah koneksi berhasil maka akan dilanjutkan untuk menjalankan 2 *sqlquery*, 1 untuk mengambil data laboratorium yang ada dan 1 untuk mengambil data video yang sudah

terunggah di dalam *database*. Daftar tersebut akan dimasukkan pada *dropdown*, dimana untuk daftar laboratorium akan berperan untuk memilih video baru akan diunggah ke laboratorium yang terpilih, dan daftar video akan digunakan dalam *dropdown* untuk memilih video manakah yang akan diperbarui. *Query* yang digunakan, dijalankan, dan diambil *array* dari data tersebut, ditunjukkan pada Potongan Kode 5-4.

```
// Ambil List Laboratorium //
$sql2 = "SELECT id_lab, nama_lab FROM laboratorium_sipil ORDER BY id_lab";
$result2 = $conn->query($sql2);

if($result2->num_rows > 0)
{
    $listLab = array();
    while($row = $result2->fetch_assoc())
    {
        //echo $row["id_lab"] . " " . $row["nama_lab"] . "<br>";
        $listLab[] = $row;
    }
}
else
{
    echo "<br>". "tidak ada daftar laboratorium pada database." . "<br>";
}

// Ambil List id dan judul untuk video
$sql3 = "SELECT id_video, judul_video, nama_lab FROM fullview_video_lab
ORDER BY id_video";
$result3 = $conn->query($sql3);
if($result3->num_rows > 0)
{
    $listVid = array();
    while($row = $result3->fetch_assoc())
    {
        $listVid[] = $row;
    }
}
```

Potongan Kode 5-4 *upload_videoLab.php* mengambil daftar laboratorium dan daftar video

Setelah daftar dari laboratorium dan video sudah didapatkan, maka akan dibuatkan *dropdown* dari daftar tersebut dengan *source code* gabungan HTML dan disisipkan pengulangan PHP yang ditunjukkan pada potongan kode 5-5, dan hasil dari kode tersebut ditampilkan pada gambar 5-5.

```

upload_videoLab.php x
<html>
<head>
<meta char-set="utf-8">
<title>Upload dan Update Video Lab</title>
</head>
<body>
<form name="formUpdateVideo" enctype="multipart/form-data" action="" method="
POST">
<label> Update Video untuk : </label>
<select name="updateVideoLab" id="uvl1">
<option value="" selected>-pilih video-</option>
<?php
foreach($listVid as $lv)
{
    $o1 = "<option value=".$lv['id_video']. ">";
    $o2 = "Video ".$lv['id_video']. " : " . $lv['judul_video'];
    $o3 = "</option>";
    echo $o1.$o2.$o3;
}
?>
</select>
<br>
<input name="updateVideo" type="file"/>
<input type="submit" value="Perbarui"/>
</form>

```

Potongan Kode 5-5 Menaruh Daftar Laboratorium dan Video pada *Dropdown*

Gambar 5-5 Tampilan halaman PHP *upload_videoLab.php*

Setelah tampilan sudah didapat dan halaman PHP tersebut sudah dapat melakukan pengunggahan data dan memberikan *value* tambahan melalui *dropdown*, maka dibuatkan kode PHP untuk mendeteksi apakah yang diunggah merupakan unggahan video baru ataukah menggantikan video yang sudah diunggah sebelumnya dalam database. Untuk awalnya akan dicek dulu apakah ada file yang terunggah dengan mengetes apakah “*count(\$_FILES)*” memiliki jumlah lebih dari 0, dan lalu mengetes apakah file yang diunggah merupakan unggahan video baru ataukah unggahan untuk memperbarui video yang sudah ada. Jika melihat Potongan Kode 5-6 dan 5-7, akan terdapat kode untuk mengambil konten dari file tersebut

menggunakan fungsi *file_get_contents()* milik PHP dan setelah itu dikonversikan sebagai *byte64* dengan menggunakan fungsi *base64_encode()*, dan setelah itu dimasukkan judul video, tipe video, dan *byte64* dari video tersebut ke dalam *database*. Ketika melakukan *update*, pada bagian *WHERE* akan otomatis mengikuti id laboratorium yang terpilih berdasarkan *dropdown* dan untuk melakukan unggah video baru, pada bagian *INSERT INTO* tidak dibutuhkan id_video karena sudah merupakan *auto-increment*.

```
// Upload Video //
if(count($_FILES) > 0)
{
    // Update Video dalam Database //
    if(isset($_FILES['updateVideo']['tmp_name']))
    {
        $namaVideo = $_FILES['updateVideo']['name'];
        $tipeVideo = $_FILES['updateVideo']['type'];
        $tmp_name = $_FILES['updateVideo']['tmp_name'];
        $idVid = $_POST["updateVideoLab"];
        if(is_uploaded_file($tmp_name))
        {
            $isi_file = file_get_contents($tmp_name);
            $vid64 = base64_encode($isi_file);

            $sqlUpdate =
            "UPDATE
            $tableName
            SET Vid_Type='$tipeVideo'
              ,Vid_64='$vid64'
            WHERE id_Video=".$_POST["updateVideoLab"];

            // jalankan query //

            if($conn->query($sqlUpdate) === TRUE)
            {
                echo "Video berhasil diperbarui";
            }
            else
            {
                echo "Video gagal diperbarui : " . $conn->error;
            }
        }
    }
    else if(isset($_FILES['uploadVideo']['tmp_name']))
    {
        // upload Video Baru ke Dalam Database //
    }
}
```

Potongan Kode 5-6 *upload_videoLab.php* memeriksa file unggahan dan kode memperbarui video dalam database

```

    }
    else if(isset($_FILES['uploadvideo']['tmp_name']))
    {
        // upload Video Baru ke Dalam Database //
        $namavideo = $_FILES['uploadvideo']['name'];
        $tipevideo = $_FILES['uploadvideo']['type'];
        $tmp_name = $_FILES['uploadvideo']['tmp_name'];
        $idlab = $_POST["uploadvideoBaru"];
        if(is_uploaded_file($tmp_name))
        {
            $sisi_file = file_get_contents($tmp_name);
            $vid64 = base64_encode($sisi_file);

            $sqlInsert =
            "INSERT INTO
            video_lab (judul_video, id_lab, Vid_64, Vid_Type)
            VALUES
            ('$namavideo', '$idlab', '$vid64', '$tipevideo')
            ";

            // jalankan query //
            if($conn->query($sqlInsert) === TRUE)
            {
                echo "Video baru berhasil diunggah";
            }
            else
            {
                echo "Video baru gagal diunggah : " . $conn->error;
            }
        }
    }
    else
    {
        echo "No files have been uploaded";
    }
    echo "<br>";
}

```

Potongan Kode 5-7 Kode *upload_videolab.php* untuk mengunggah video baru ke dalam database

Saat melakukan pengunggahan data dalam ukuran besar, perubahan pada file ‘php.ini’ dan ‘my.ini’ akan perlu dilakukan perubahan besar *value*. Di dalam *file php.ini*, *post_max_size* dan *upload_max_filesize* dapat diubah sehingga memiliki ukuran file maksimal yang dapat diunggah. Untuk pada *file my.ini*, nilai *max_allowed_packet* dapat diubah terlebih dahulu supaya pada saat pengunggahan *file* dengan ukuran besar tidak terjadi error ‘*MYSQL Server has gone away*’, dan perubahan pada *innodb_log_file_size* yang perlu diberikan *value* lebih besar sepuluh kali lipat daripada ukuran *file* maksimal yang ingin diunggah.

5.2.4.2 Pembuatan Halaman Pengambil Daftar Video

Dalam aplikasi, daftar video akan ditunjukkan dalam sebuah antarmuka, dan daftar video tersebut akan diambil dari *database*. *Unity* tidak mampu untuk mengambil *array* secara langsung dari *database*, sehingga dibutuhkan halaman *web* sebagai perantara untuk mengambil data *array* tersebut, dan diambil dalam bentuk *JSON string*, sehingga akan lebih mudah untuk diambil dan dibaca oleh *Unity*. Halaman PHP yang akan dibuat dalam tahap ini diberi nama '*json_videolist.php*'. Untuk mengambil data dari *database* dan dirubah menjadi JSON menggunakan PHP cukup sederhana.

```

1  <?php
2  // json_videolist.php untuk Lab//
3  $servername = "localhost";
4  $username = "root";
5  $password = "";
6  $dbname = "db_unity";
7  $tableName = "no64_videolab"; // ini view
8  $sql = "SELECT * FROM $tableName";
9
10 $conn = mysqli_connect($servername, $username, $password, $dbname);
11
12 if($conn->connect_error)
13 {
14     die("Connection Failed : " . $conn->connect_error);
15 }
16
17 // Fetch Table //
18 if(isset($_GET['idLab']))
19 {
20     $idLab = $_GET['idLab'];
21     // $sql = "SELECT * FROM $tableName WHERE id_Lab = $idLab";
22     $sql = $sql." WHERE id_lab = $idLab ORDER BY id_video";
23 }
24 else
25     $sql = $sql." ORDER BY id_lab";
26
27 $result = $conn->query($sql);
28 $semparray = array();
29 while($row = $result->fetch_assoc())
30 {
31     $semparray[] = $row;
32 }
33
34 echo json_encode($semparray);
35 ?>

```

Potongan Kode 5-8 Source Code untuk *json_videolist.php*

Awal pembuatan kodenya adalah diawali dengan melakukan koneksi ke database dan dilakukan pengambilan data, data yang diambil akan ditaruh di dalam sebuah *array*, dan lalu cukup panggil dan lakukan *encoding* pada *array* tersebut dengan *json_encode()* untuk langsung mendapatkan data *array* dalam bentuk JSON. Sebelum pengambilan data dilakukan, akan dilakukan cek terlebih dahulu apakah ada nilai untuk *\$_GET*, karena salah satu fokus tujuan dari halaman PHP ini adalah untuk juga dapat mengambil daftar video berdasarkan ID dari laboratorium tertentu. Jika tidak ada value *\$_GET* maka akan mengambil seluruh data video tanpa melihat laboratorium. Untuk kode yang dibuat dapat dilihat pada potongan kode 5-8.

5.2.4.3 Pembuatan Halaman Menampilkan Video

Video yang telah diunggah ke dalam *database* tetap akan perlu diambil. *Unity* hanya dapat mengambil video menggunakan fitur WWW apabila konten halaman yang diambil adalah video, sehingga perlu dibuatkan halaman PHP untuk menampilkan sebuah video dari *database* yang diawali dengan *header()* yang dengan *Content-Type* bertipe video, karena apabila video tersebut ditampilkan dengan tanpa cara tersebut seperti dimisalkan ditampilkan saja menggunakan *video player* milik *HTML*, maka *Unity* tidak dapat mengambilnya. Untuk penamaan, halaman PHP ini akan dinamakan sebagai '*display_videoLab.php*'. Dalam pembuatannya, awalnya perlu diperiksa terlebih dahulu apakah ada nilai untuk ID video dalam *\$_GET*, karena halaman ditujukan untuk mengambil hanya satu video. Selanjutnya sama dengan halaman-halaman PHP sebelumnya, yakni dilakukan koneksi ke *database* dan diberikan *query* untuk data yang akan diambil. Saat pengambilan data berlangsung, akan langsung diambil tipe video dan langsung dilakukan *decoding* untuk video yang masih dalam bentuk *byte64* saat dilakukan pengambilan data. Dalam pengambilan data tidak dibutuhkan *array* untuk menyimpan data-data video yang diambil, karena data video yang diambil pastinya hanya 1 baris. Setelah data tersebut sudah didapatkan, selanjutnya adalah untuk menampilkannya video tersebut.


```

1  <?php
2  // Display video Lab //
3  if(isset($_GET['idVid']))
4  {
5      $servername = "localhost";
6      $username = "root";
7      $password = "";
8      $dbname = "db UNITY";
9      // Create Connection //
10     $conn = new mysqli($servername, $username, $password, $dbname);
11     // Check Connection //
12     if($conn->connect_error)
13     {
14         die("Connection failed : " . $conn->connect_error);
15     }
16
17     $tableName = "fullview_video_lab";
18     $sql = "SELECT Vid_Type, Vid_64 FROM $tableName WHERE id_video = " . $_GET['idVid'];
19     $result = $conn->query($sql);
20
21     // Ambil Data //
22     if($result->num_rows > 0)
23     {
24         while($row = $result->fetch_assoc())
25         {
26             $tipeVid = $row['Vid_Type'];
27             $ekstensi = explode("/", $tipeVid);
28             $v64 = base64_decode($row['Vid_64']);
29         }
30     }
31     // Displaying Video Stuff //
32     $file = 'temporaryVideos/videoLab' . $_GET['idVid'] . "." . $ekstensi[1];
33     file_put_contents($file, $v64);
34     if(file_exists($file))
35     {
36         header('Content-Type: ' . $tipeVid);
37         readfile($file);
38     }
39 }
40 else
41 {
42     echo "tidak ada id video";
43 }
44 ?>

```

Potongan Kode 5-9 Kode PHP untuk *display_videoLab.php*

Untuk dapat menampilkan video tersebut dan langsung menjadikan sebuah *link URL* dalam bentuk tipe video terkait, maka ditaruh dulu file tersebut di dalam *host* dengan menggunakan *file_put_contents()* yang diisikan tempat dimana video tersebut akan diletakkan yang mana namanya akan mengikuti ID dari video yang dipanggil dan hasil *decoding* dari *byte64*. Perhatikan dalam penamaan *file* tetap dibutuhkan ekstensi, maka dari itu saat *file* diunggah dalam *database* dimasukkan tipe data dari *file* yang diunggah. Tahapan selanjutnya baru akan berjalan setelah *file_put_contents()* sudah selesai menaruh file. Jika file sudah ada, maka halaman akan dibuat sebagai *file* tipe video terkait menggunakan *header()* dan

pemanggilan *file* video dilakukan dengan menggunakan *readfile()*. Untuk *source code* PHP secara penuh dapat dilihat pada potongan kode 5-9

5.2.5 Pembuatan Aset Aplikasi

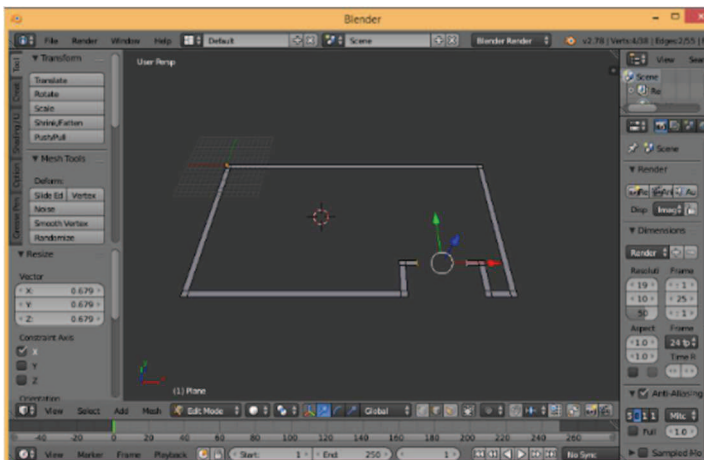
Berikut adalah bahasan bagaimana aset-aset untuk diletakkan di dalam folder *asset* pada *Unity Project* untuk pembuatan aplikasi dibuat.

5.2.5.1 Pembuatan Map

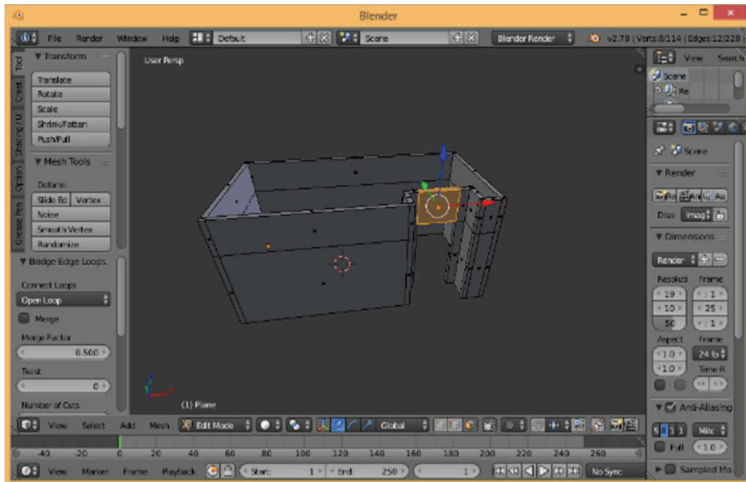
Pembuatan asset untuk *scene* peta Unity dimulai pada bagian ini.

5.2.5.1.1 Pembuatan Objek Tiga Dimensi

Untuk awal pembuatan gedung menggunakan blender, awalnya akan dibuat denah-nya terlebih dahulu menggunakan objek dari *Blender*. Setelah itu denah tersebut dapat ditarik kea rah atas untuk membuat dinding dengan menggunakan fitur *extrude* pada Blender. Dalam pengerjaan model gedung dalam penelitian ini, *extrude* dilakukan sekian kali, menyesuaikan dengan ukuran dan lokasi dari pintu dan jendela, sehingga dapat mempermudah pembuatan lubang untuk pintu maupun jendela.



Gambar 5-6 Membuat Denah menggunakan *Blender*

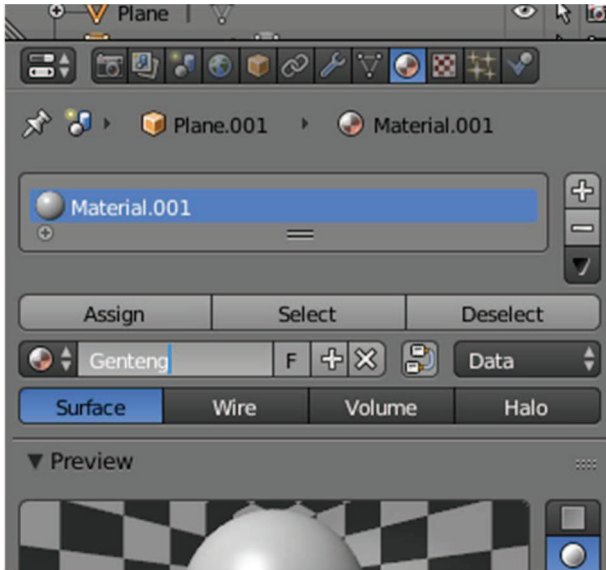


Gambar 5-7 Meningkatkan Denah sebagai Dinding menggunakan fitur *Extrude* pada *Blender*

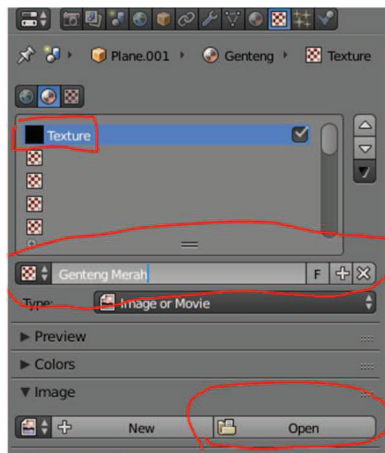
Setelah didapatkan dinding, maka bisa dilanjutkan dengan pembuatan lantai dengan melakukan sejumlah *extrude* pada sebuah plane dan untuk jendela bisa dilakukan dengan memilih *face* tertentu

5.2.5.1.2 Pemberian dan Pengaturan untuk Material dan Tekstur

Untuk pemberian dan pengaturan tekstur awalnya akan dilakukan pada program *Blender* terlebih dahulu supaya lebih mudah. Untuk pertama-tama perlu dicari terlebih dahulu tekstur yang diinginkan. Untuk tekstur, bisa dicari melalui pencarian gambar *Google*, dan usahakan untuk menggunakan tekstur dengan ukuran yang tidak terlalu besar karena ukuran tekstur yang besar tersebut dapat mempengaruhi performa dari aplikasi.



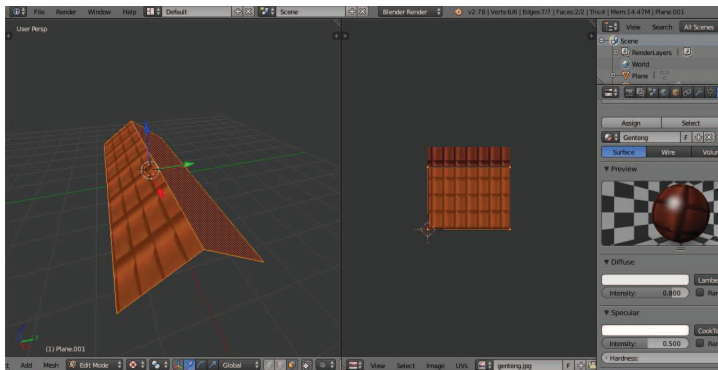
Gambar 5-8 Buat Material dan baru



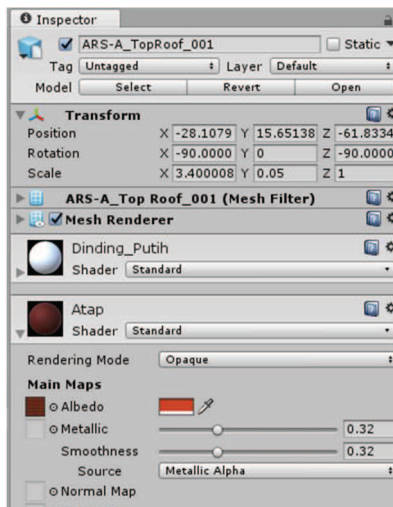
Gambar 5-9 Buat Tekstur Baru

Selanjutnya untuk dapat melihat apakah tekstur sudah masuk, pastikan *viewport* dipilih mode *Texture*. Jika material atau

tekstur terlihat putih, dimana tidak sama dengan gambar tekstur yang diunggah, maka perlu masuk kembali kepada edit mode, pilih face yang bermasalah, dan dilakukan *UV Unwrap*, dan lalu ubah ukuran tekstur sedemikian rupa hingga sesuai pada mode *UV/Image Editor*.



Gambar 5-10 (Kiri) Edit Mode dengan *Viewport Texture* dan (Kanan) *UV/Image Editor*



Gambar 5-11 Pengaturan Material dan Tekstur pada Unity

Selanjutnya begitu objek tersebut sudah dimasukkan ke dalam *Asset* milik *Unity*, tekstur yang digunakan dalam Blender perlu dimasukkan juga ke dalam asset *Unity*. Pilih objek yang memiliki material tersebut, dan lalu lakukan *drag and drop* gambar tekstur yang dipakai ke bagian '*Albedo*'. Selanjutnya besar kecilnya *Metallic* dan *Smoothness* dapat diatur daya material untuk memantulkan cahaya supaya lebih sesuai dengan aslinya.

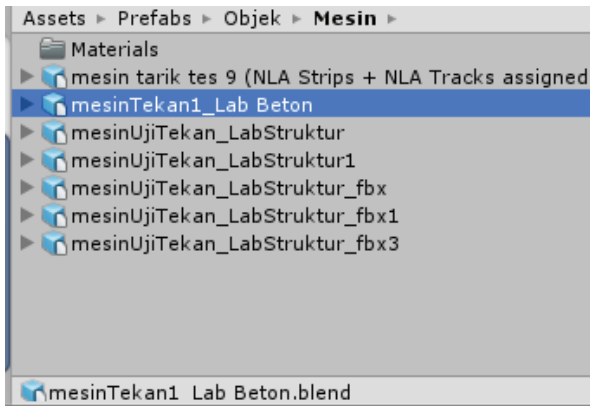
5.2.5.1.3 Pembuatan *Scene*

Scene merupakan *save file* yang akan berfungsi menyimpan posisi objek apa saja yang telah ditaruh di dalam editor. Dalam sebuah *video game*, *scene* ini dapat dikatakan sebagai sebuah *level*, sehingga file tersebut salah satunya bisa dipakai jika ingin suatu saat merubah tempat secara keseluruhan namun tetap dalam satu aplikasi. Jika ingin membuat tempat yang berbeda dengan posisi objek-objek yang berbeda maka dapat membuat *scene* baru sehingga posisi objek-objek pada bagian sebelumnya masih bisa disimpan untuk dipanggil lagi.

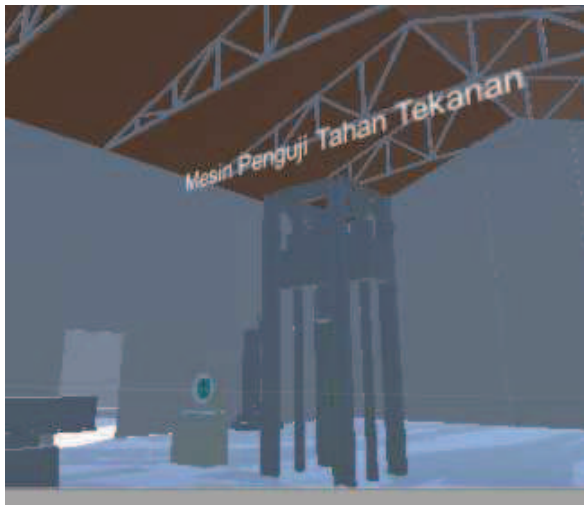
Untuk pembuatan *scene*, antarmuka pengguna bisa dibuat menggunakan *Unity Editor*. Untuk objek-objeknya, apabila sudah diekspor oleh Blender dan dimasukkan ke dalam folder *Asset* dalam *Unity*, maka *file* dengan ekstensi *.blend* ataupun *.fbx* tersebut dapat dilakukan *drag-and-drop* melalui *Unity Editor* dan diposisisikan sedemikian rupa. Untuk Objek-objek tersebut, supaya tidak bisa ditembus oleh karakter, pastikan *Generate Colliders* sudah dicentangkan. *Generate Colliders* dapat ditemukan ketika *prefab* objek tersebut di klik, lalu dalam *inspector* pindah ke *tab Model*.

5.2.5.1.4 Peletakan Tanda Interaksi

Untuk interaksi di dalam simulasi akan diberikan tanda bagian mana saja dapat dilakukan interaksi, karena tidak semua objek memiliki interaksi masing-masing, sehingga perlu dibedakan. Salah satu tanda yang bisa diberikan kepada objek adalah dengan memberikannya tanda penunjuk seperti tanda panah atau teks.



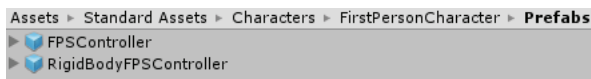
Gambar 5-12 Prefab untuk Mesin



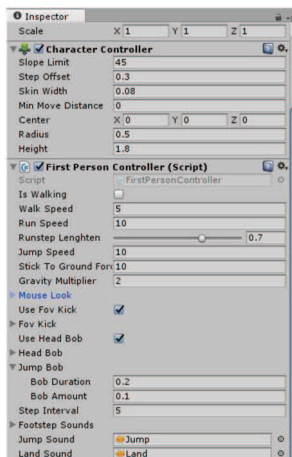
Gambar 5-13 Penanda Objek

5.2.5.2 Pengontrol Karakter

Untuk pengontrol karakter, dalam pengembangan ini digunakan *FirstPersonCharacter* yang disediakan dalam *Standard Asset* milik Unity. Import dahulu *Standard Asset* yang menyimpan data-data *FirstPersonCharacter*, dan selanjutnya cukup *drag-and-drop prefab* dengan nama *FPSController* dan pastikan *main camera* cukup satu saja saat *game* dijalankan supaya tidak terjadi *error* atau *warning* karena dideteksi terdapat lebih dari 1 kamera. Untuk pengaturan kontrol, *FPS Controller* ini memiliki beberapa variabel yang bisa diatur sedemikian rupa untuk menyesuaikan dengan linkup yang telah dibuat seperti tinggi kamera, kecepatan jalan, dan sebagainya, ditunjukkan pada gambar 5-15.



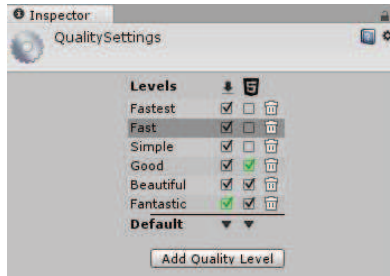
Gambar 5-14 Prefab FPSController



Gambar 5-15 Public Value yang dapat diubah diubah pada FPSController dari StandardAsset

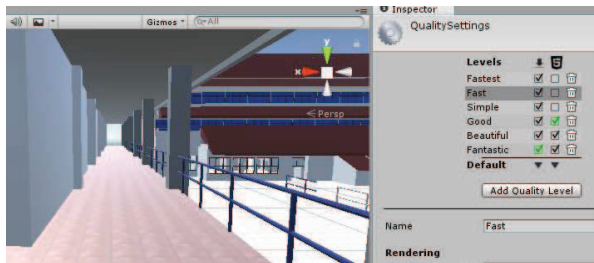
5.2.5.3 Pengaturan *Unity Project*

Dalam pengerjaan proyek dalam *Unity Editor*, *Unity* menyediakan laman untuk mengatur kualitas grafis dari proyek aplikasi. Untuk dapat menentukan kualitas dari peta 3D, dapat dilihat dalam *Unity Editor* pada menu **Edit > Project Settings > Quality**.



Gambar 5-16 Pengaturan Kualitas *Unity*

Dalam pengerjaan penelitian ini, akan digunakan kualitas tingkat *good*. Hal ini dikarenakan pada tingkat *Fastest* hingga *Simple*, bayangan dari objek tidak ditampilkan dengan baik, dan penerapan tampilan bayangan baru mulai terlihat pada kualitas *Good*. Pada kualitas *Beautiful* dan *Fantastic* tentunya memiliki kualitas grafis yang lebih baik, namun melihat aplikasi ini merupakan aplikasi web, maka tidak diberikan kualitas grafis yang terlalu bagus karena dikhawatirkan akan memberatkan kinerja perangkat yang menjalankannya.

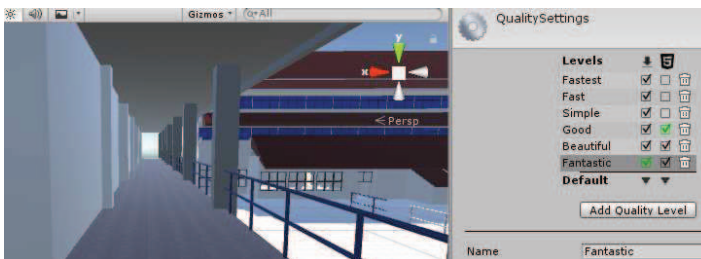


Gambar 5-17 Tampilan kualitas *Fast*



Gambar 5-18 Tampilan Kualitas *Good*

Dalam kualitas *Good* bayangan sudah terimplementasikan, namun tidak untuk bayangan pada objek jarak jauh, dimana bayangan tersebut baru akan muncul ketika kamera sudah dekat dengan objek tersebut. Bayangan dari objek jauh tersebut akan tetap muncul apabila menggunakan kualitas *beautiful* atau *fantastic*, namun hal tersebut berarti akan melakukan *rendering* yang lebih dimana akan memakan *memory* lebih juga.



Gambar 5-19 Tampilan Kualitas *Fantastic*

5.2.5.4 Pembuatan Menu Aplikasi

Dalam aplikasi ini akan terdapat beberapa menu untuk membantu menampilkan penjelasan dari tempat ataupun suatu mesin. Untuk UI sendiri akan dibuat melalui Unity Editor dan selanjutnya diberikan script untuk memanggilnya.

5.2.5.4.1 Pembuatan *Main Menu* pada Aplikasi

Main menu adalah tampilan antarmuka pengguna yang ditampilkan pada bagian awal aplikasi, dimana pengguna umum akan dapat memulai untuk menjelajahi peta tiga dimensi ataupun melihat penjelasan mengenai aplikasi.

Untuk mengatur apa yang akan terjadi ketika button di-klik, maka akan dibuatkan kode yang ditunjukkan pada Potongan Kode 5-10. Dari kode tersebut, cukup ditaruh pada *canvas*, *drag-and-drop* panel terkait melalui *editor*, dan pada *drag-and-drop* juga script tersebut untuk *button* masing-masing sehingga ketika klik terjadi pada *button* akan menjalankan *function* yang telah diatur.

```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class mainMenu : MonoBehaviour {
6      // drag and drop panel terkait lewat editor
7      public GameObject panel_mainMenu;
8      public GameObject panel_menuAbout;
9
10     public void klikMulai()
11     {
12         Application.LoadLevel("sceneUtama");
13     }
14
15     public void klikTentang()
16     {
17         panel_mainMenu.SetActive(false);
18         panel_menuAbout.SetActive(true);
19     }
20
21     public void klikKembali()
22     {
23         panel_mainMenu.SetActive(true);
24         panel_menuAbout.SetActive(false);
25     }
26 }

```

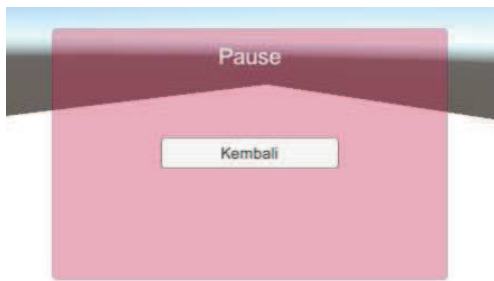
Potongan Kode 5-10 *mainMenu.cs*



Gambar 5-20 Menu Utama Aplikasi

5.2.5.4.2 Pembuatan *Pause Menu*

Pause menu ditampilkan ketika pengguna sedang dalam aplikasi dan membuka menu untuk menghentikan aplikasi sementara sehingga dapat mengakses menu atau informasi lain pada saat itu yang bukan merupakan interaksi dengan objek. Untuk kode dari *pause menu* ditunjukkan pada potongan kode 5-11. Di dalam potongan kode tersebut, objek untuk *kanvasMenu* dan *Player* masih perlu diatur lagi dalam editor. Adapun kode `GetComponent<FirstPersonController>()` adalah untuk mengambil script *FirstPersonController* yang ada dalam *scene*, dan fungsi *enabled* dipanggil untuk membuat *script* tersebut aktif atau tidak, sehingga memungkinkan karakter tidak bisa digerakkan ketika *pause menu* sedang aktif.



Gambar 5-21 *Pause Menu*

```

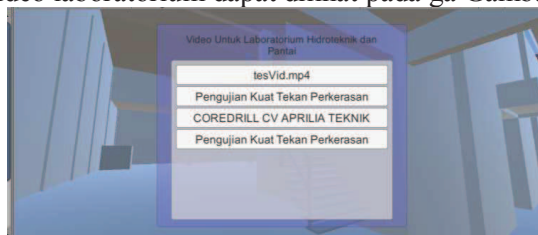
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityStandardAssets.Characters.FirstPerson;
5  public class PauseScreen : MonoBehaviour {
6      public GameObject kanvasMenu;
7      public GameObject Player;
8      bool pausing = false;
9      void Start () {
10         kanvasMenu.SetActive (false);
11         pausing = false;
12     }
13     void Update () {
14         if (Input.GetKey(KeyCode.Escape)){
15             pausing = !pausing;
16             ispaused();
17         }
18     }
19     public void ispaused(){
20         if (pausing==true) {
21             Player.GetComponent<FirstPersonController> ().enabled = false;
22             Time.timeScale = 0;
23             kanvasMenu.SetActive (true);
24         } else if(pausing==false) {
25             Player.GetComponent<FirstPersonController> ().enabled = true;
26             Time.timeScale = 1;
27             kanvasMenu.SetActive (false);
28         }
29     }
30 }
31

```

Potongan Kode 5-11 *pauseScreen.cs*

5.2.5.4.3 Pembuatan *Menu* untuk memilih video pada laboratorium

Menu ini akan tampil ketika pengguna sedang berada pada area tertentu dan menekan tombol interaksi. Pada menu tersebut akan ditampilkan video apa saja yang ada mengenai lab tersebut. Untuk dapat melakukan bagian ini juga diperlukan tambahan *LitJson.dll* ke dalam *library*, sehingga Unity dapat membaca data *Json* dan merubahnya menjadi *array* yang dapat dibaca dan diambil datanya di dalam *Unity*. Tampilan dari daftar video laboratorium dapat dilihat pada Gambar 5-22.



Gambar 5-22 Antarmuka Daftar Laboratorium

Dalam pembuatan daftar video dan untuk menampilkan video tersebut ketika terjadi klik pada *button*, akan dibutuhkan 4 *script*, yakni *showCanvas_ListVideo.cs*, *listVideoLab.cs*, *buttonPlayVidLab.cs*, dan *videoDisplayer.cs*.

5.2.5.4.3.1 *showCanvas_listVideo.cs*

Tahapan jalan dari *script* tersebut akan dimulai dari *showCanvas_listVideo.cs*. Pada bagian awal *showCanvas_listVideo.cs*, akan dibuatkan *public variable* terlebih dahulu, dimana *string* 'namaCanvasClone' dipakai untuk mendeteksi apakah antarmuka untuk daftar video sedang ditampilkan (nama, dan terdapat '*public listVideoLab prefab_canvasListVideo*' yang akan digunakan untuk menyimpan prefab *canvas* yang akan menunjukkan daftar video dari lab terkait. Selanjutnya dalam *function public void Update()* akan terus diperiksa setiap frame, apakah '*player*' sedang berada di dalam *trigger area*. Untuk mengecek keberadaan *game object* dengan tag '*player*', perlu diberikan *onTriggerEnter()* untuk memeriksa apakah sedang ada *player* yang masuk saat itu dan *onTriggerExit()* untuk mengecek apakah *player* sudah keluar dari *trigger area*. Sebelumnya pastikan bahwa *FPS Controller* sudah diberikan tag '*player*'. Kode *script* dari *showCanvas_listVideo.cs* akan ditunjukkan pada Potongan Kode 5-12.

Untuk peletakan *script showCanvas_listVideo.cs* ini, akan ditaruh pada *empty game object* dengan diberikan *box collider* yang telah diberikan ukuran besar yang sesuai dan telah diatur sebagai *isTrigger*, sehingga menjadikannya sebuah *trigger area* untuk mendeteksi masuk keluarnya *player*. Untuk membedakan *trigger area* ini bagi masing-masing laboratorium, isi dari variabel *prefab_canvasListVideo* akan berbeda-beda berdasarkan laboratorium terkait. Untuk penggambarannya akan ditunjukkan pada Gambar 5-23.

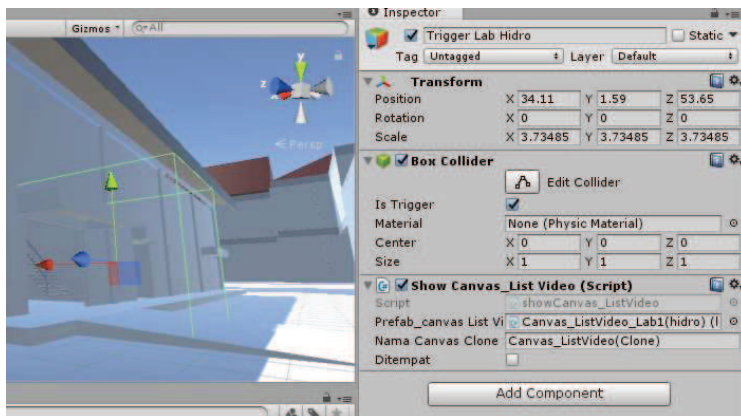
```

Teksip_Prototype-2
showCanvas_ListVideo

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4
5  public class showCanvas_ListVideo : MonoBehaviour {
6      public listVideoLab prefab_canvasListVideo;
7      public string namaCanvasClone = "Canvas_ListVideo(Clone)";
8      public bool ditempat = false;
9
10     public void Start()
11     {
12         Cursor.visible = true;
13     }
14     public void Update()
15     {
16         if(ditempat)
17         {
18             if(!GameObject.Find(namaCanvasClone))
19             {
20                 if(Input.GetKeyDown(KeyCode.E))
21                 {
22                     listVideoLab canvasListVideo = (listVideoLab)Instantiate(prefab_canvasListVideo);
23                     Cursor.visible = true;
24                 }
25             }
26             //else{if(GameObject.Find(namaCanvasClone)) Destroy(GameObject.Find(namaCanvasClone))}
27         }
28     }
29     void OnTriggerEnter(Collider pemicu)
30     {
31         if(pemicu.gameObject.tag == "Player")
32             ditempat = true;
33     }
34     void OnTriggerExit(Collider pemicu)
35     {
36         if(pemicu.gameObject.tag == "Player")
37             ditempat = false;
38     }
39
40 }

```

Potongan Kode 5-12 *showCanvas_listVideo.cs*



Gambar 5-23 *Trigger Area* untuk Laboratoirum Hidroteknik

5.2.5.4.3.2 *listVideoLab.cs*

Dalam interaksi *showCanvas_ListVideo.cs* akan terjadi *Instantiate* untuk memanggil *prefab* yang memiliki komponen *script listVideoLab.cs*. Hal tersebut memiliki arti bahwa antarmuka pengguna yang sebelumnya dibuat pada desain *menu* akan dipanggil. Untuk pembuatan *script listVideoLab.cs*, akan diperlukan library *LitJson* yang bisa diunduh dari *GitHub*, dimana *LitJson* akan digunakan untuk membaca data JSON yang diambil dari URL menggunakan WWW. Untuk bagian awal dari *script* ditunjukkan pada Potongan Kode 5-13. *ListJson.dll* yang sudah diunduh dari *GitHub* jika sudah ditaruh dalam folder *Assets* milik proyek *Unity*, maka sudah dapat langsung dipanggil menggunakan *using* di awal *script*. Dalam Potongan Kode 5-13 sudah diberikan komentar untuk menandai mana variabel yang berhubungan dengan antarmuka dan variabel mana yang berhubungan dengan *database*. Untuk variabel antarmuka akan mencakup *panelMain* sebagai *background* dan *panel* yang akan membawa *prefab_buttonVid* yang terpanggil berdasarkan *database*. Untuk bagian *database*, diperlukan tautan halaman PHP untuk mengambil data JSON dari *database* yang sudah dibuat sebelumnya. Untuk variabel 'varGet' akan diisi variabel *\$_GET* yang sesuai dengan apa yang dibutuhkan halaman PHP tersebut, dimana berdasarkan halaman PHP yang dibuat sebelumnya, variabel ini adalah 'idLab'. Tidak lupa diberikan '=' pada bagian akhir dan untuk isi dari variabel tersebut nantinya, disimpan pada variabel 'valGet' dalam *script* ini. Nilai dari *valGet* akan mengikuti ID dari laboratorium terkait, yang mana bisa diganti nilainya melalui *Unity Editor* untuk laboratorium yang berbeda-beda, sehingga jika *script* ini dipakai untuk menunjukkan video laboratorium hidroteknik, maka *valGet* adalah 1, untuk laboratorium struktur, masukkan 5 pada *valGet*, dan seterusnya.


```

eksip_Prototype-2 - listVideoLab
1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5  using LitJson;
6
7  public class listVideoLab : MonoBehaviour {
8      // UI's Variables //
9      public Transform panelMain;
10     public Transform panelVidLister;
11     public buttonPlayVidLab prefab_buttonVid;
12     public Text judulPanel;
13     public string lokasiJudulPanel = "text_judul";
14     public string b; // String depan judul
15     // db stuff's Variables //
16     public string urlJsonVidList = "http://localhost/unityweb/json_videolist.php?";
17     public string varGet = "idLab=";
18     public string valGet = "3"; //masukkan id lab yang diinginkan di sini. Silahkan dig
19     public string namaChildVideo = "videoDisplay(Clone)"; // coba pakai ini in case nam
20     public void Start()
21     {
22
23         judulPanel = GameObject.Find(lokasiJudulPanel).GetComponentInChildren<Text>();
24         StartCoroutine(ambilJson());
25     }

```

Potongan Kode 5-13 *listVideoLab.cs* bagian 1: Variabel dan *Start()*

Dalam mulainya proses *script* ini, akan dimulai dengan mengambil komponen teks judul panel dan memanggil fungsi *ambilJson()*. Fungsi *ambilJson()* diawali dengan menggunakan fitur WWW. Untuk URL yang digunakan adalah dengan mengambil variabel yang sudah dibuat sebelumnya, yakni dengan menggabungkan *urlJsonVidList*, *varGet*, dan *valGet*, sehingga didapatkan tautan penuh untuk memanggil halaman PHP *json_videolist.php* beserta *\$_GET* sehingga bisa mendapatkan daftar video dari laboratorium tertentu. Setelah isi dari URL diambil, akan langsung dijadikan sebagai *array JsonData* menggunakan *JsonMapper.ToObject()*, dan dilanjutkan dengan perulangan untuk memanggil *button*, mengatur *parent* dari *button* tersebut, mengganti teks dari *button* tersebut sehingga sama dengan daftar video pada perulangan saat itu, memberikan info ID dari video tersebut untuk dipanggil nantinya saat ingin memainkan video, dan mengatur tampilan nama laboratorium sebagai judul dari panel nantinya. Untuk fungsi *Update()*, akan diatur jika tombol *esc* ditekan saat tidak ada video yang sedang dimainkan, maka antarmuka daftar video tersebut akan ditutup. Untuk *script* yang

mencakup bagian *Update()* dan *ambilJson()* akan ditunjukkan pada Potongan Kode 5-14.

```

26 public void Update()
27 {
28     if(Input.GetKeyDown(KeyCode.Escape))
29     {
30         if(this.transform.Find(namachildvideo))
31         {}
32         else
33         {
34             Destroy(this.gameObject);
35             Cursor.visible = false;
36         }
37     }
38 }
39 public IEnumerator ambilJson()
40 {
41     WWW getJson = new WWW(urlJsonVidList + varGet + valGet);
42     yield return getJson;
43     JsonData listVideo;
44     listVideo = JsonSerializer.ToObject(getJson.text);
45     string a = ""; // string akhir judul
46     for(int i=0; i<listVideo.Count; i++)
47     {
48         // Panggil prefab buttonPlayVidLab di sini
49         buttonPlayVidLab buttons = (buttonPlayVidLab)Instantiate(prefab_buttonVid);
50         buttons.transform.SetParent(panelVidListler, false);
51         buttons.buttonText.text = (string) listVideo[i]["judul_video"];
52         buttons.valGet = (string) listVideo[i]["id_video"];
53         a = (string) listVideo[i]["nama_lab"];
54     }
55     judulPanel.text = b+a;
56 }
57

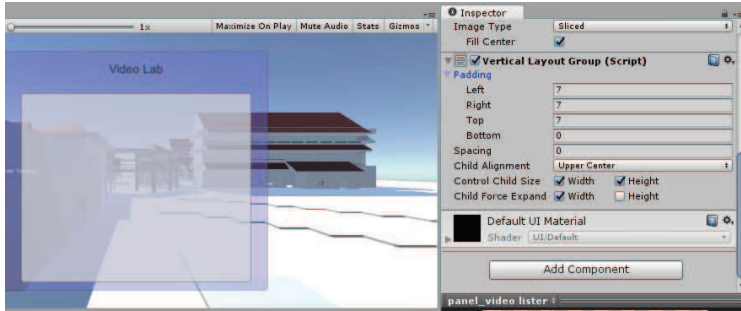
```

Potongan Kode 5-14 *listVideoLab.cs* bagian 2: *Update()* dan *ambilJson()*

Untuk objek yang akan menjadi *prefab* bagi *listVideoLab.cs* adalah *canvas*, *panel*, *teks judul*, dan *panel* yang akan menyimpan daftar video, seperti yang ditunjukkan pada Gambar 5-24. Pastikan juga bagian *panel* yang akan menyimpan daftar video (*panel_videoLister*) diberikan komponen *Vertical Layout Group* dengan isi seperti pada Gambar 5-25 sehingga saat *Instantiate prefab button* akan tersusun rapi berdasarkan *panel* tersebut.



Gambar 5-24 Objek yang akan Menjadi *Prefab* bagi *listVideoLab*




Gambar 5-25 Tampilan Antarmuka untuk *Prefab* (Kiri) dan Komponen *Vertical Layout Group* (Kanan)

5.2.5.4.3.3 *buttonPlayVid.cs*

Script buttonPlayVid.cs akan berfungsi untuk menyimpan ID dan judul dari masing-masing video terkait yang telah dipanggil oleh *listVideoLab.cs*, dan berfungsi juga untuk menyimpan fungsi apa yang akan terjadi ketika dilakukan klik pada *button*. Sama seperti *listVideoLab.cs*, *script* ini juga akan memiliki URL yang merujuk pada *display_videoLab.php*, dengan *string variable* *varGet* dan *valGet* yang masing-masing merupakan *\$_GET* dan nilainya, yang mana jika ketiga variabel tersebut digabung dan dipanggil, akan merujuk kepada video dengan ID terkait. Nilai dari *valGet* akan mengikuti apa yang dipanggil pada *listVideoLab.cs* sebelumnya, karena dalam Potongan Kode 5-14 terlihat bahwa terjadi perubahan pada *'button.valGet'*. Dalam *buttonPlayVidLab.cs*, fungsi yang dimiliki hanyalah *ShowVideo()*, dimana fungsi tersebut akan berjalan ketika terjadi klik pada *button* tersebut, yang mana nanti akan diatur dalam *Unity Editor*. Ketika *ShowVideo()* dijalankan, akan dimunculkan *RawImage* yang berperan sebagai *video player* menggunakan *script videoDisplayer.cs*, dimana *urlVideo* dari *videoDisplayer.cs* akan diubah dengan tautan berupa gabungan dari *incVidLink*, *varGet*, dan *valGet*, dan lalu memerintahkan fungsi *mulai()* dalam *script videoDisplayer.cs*. Untuk peletakan kompenen *script* ini, cukup diletakkan pada antarmuka *button* dan dijadikan *prefab*. *Script*

lengkap dari *buttonVidLab.cs* ditunjukkan pada Potongan Kode 5-15



```

1  using System.Collections;
2  using System.Collections.Generic;
3  using UnityEngine;
4  using UnityEngine.UI;
5
6  public class buttonPlayVidLab : MonoBehaviour {
7      public Transform panelMain;
8      public Transform panelButton;
9      public Text buttonText;
10     public string incvidLink = "http://localhost/unityweb/display_videoLab.php?";
11     public string varGet = "idVid=";
12     public string valGet = "1";
13     public videoDisplayer prefab_videoDisplayer;
14
15     public void ShowVideo()
16     {
17         videoDisplayer displayVid = (videoDisplayer)Instantiate(prefab_videoDisplayer);
18         displayVid.transform.SetParent(this.transform.root, false);
19         displayVid.urlVideo = incvidLink+varGet+valGet;
20         displayVid.mulai();
21     }
22 }

```

Potongan Kode 5-15 *buttonPlayVidLab.cs*

5.2.5.4.3.4 *videoDisplayer.cs*

Script terakhir dalam pembuatan daftar video laboratorium ini akan berperan untuk menampilkan video. *Script* ini akan ditaruh pada sebuah *RawImage* yang lalu dijadikan *prefab*. Pastikan *RawImage* tersebut sudah diberikan komponen *VideoPlayer* dan *AudioSource* terlebih dahulu sebelum dijadikan *prefab*. Patut diperhatikan pada bagian awal *script* perlu ada *using UnityEngine.Video* dan *UnityEngine.UI* seperti yang ditunjukkan pada Potongan Kode 5-16. Pada potongan kode tersebut juga terdapat variabel *urlVideo*, dimana nilai dari variabel tersebut sudah dirubah melalui kode yang ditunjukkan pada Potongan Kode 5-15. Dalam fungsi *Update()*, akan disisikan apabila tombol *esc* ditekan, maka video akan ditutup dan kembali ke antarmuka daftar video. Pada fungsi *mulai()* akan diatur isi dari beberapa variabel, dimana merupakan variabel yang ada dalam objek itu sendiri, sehingga tidak perlu repot-repot memasukkannya secara *drag-and-drop* pada *Unity Editor*.

```

Teksip_Prototype-2
1 using System.Collections;
2 using System.Collections.Generic;
3 using UnityEngine;
4 using UnityEngine.UI;
5 using UnityEngine.Video;
6
7 public class videoDisplayer : MonoBehaviour {
8     public RawImage displayer;
9     public VideoPlayer vidPlayer;
10    private VideoSource vidSource;
11    public AudioSource audSource;
12    public string urlVideo = "http://localhost/unityweb/showVideo.php";
13
14    public void Update()
15    {
16        if(Input.GetKeyDown(KeyCode.Escape))
17        {
18            Destroy(this.gameObject);
19        }
20    }
21
22    public void mulai()
23    {
24        displayer = this.GetComponent<RawImage>();
25        vidPlayer = this.GetComponent<VideoPlayer>();
26        audSource = this.GetComponent<AudioSource>();
27        Application.runInBackground = true;
28        StartCoroutine(playVid());
29    }

```

Potongan Kode 5-16 *videoDisplayer.cs* Bagian 1: Variabel, *Update()*, dan *mulai()*

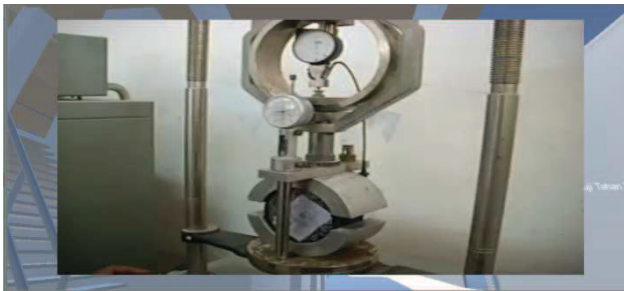
Untuk dapat memainkan video, kode yang digunakan ada di dalam fungsi *playVid()* sebagaimana ditunjukkan pada Potongan Kode 5-17. Untuk URL video terkait tidak dipanggil menggunakan WWW, namun menggunakan fitur *VideoPlayer.url*, dimana URL video terkait sudah didapatkan bersamaan dengan pengambilan daftar video. Fitur ini masih terbilang baru pada *Unity 5.6*, dimana komponen *video player* baru muncul pada versi tersebut, sehingga dapat membantu memendekkan kode dari *script*. *Video player* sebagai *texture* pada *RawImage* juga dapat terbilang lebih ringan daripada menggunakan *movie.texture*.

```

IEnumerator playvid()
{
    // Disable Play on Awake for both video and Audio
    vidPlayer.playOnAwake = false;
    audSource.playOnAwake = false;
    audSource.Pause();
    // Play video from url
    vidPlayer.source = VideoSource.Url;
    vidPlayer.url = urlVideo;
    vidPlayer.audioOutputMode = VideoAudioOutputMode.AudioSource;
    vidPlayer.EnableAudioTrack(0, true);
    vidPlayer.SetTargetAudioSource(0, audSource);
    // Set video to play then prepare audio to prevent buffering
    vidPlayer.Prepare();
    // Wait until video is prepared
    WaitForSeconds waitTime = new WaitForSeconds(1);
    while (!vidPlayer.isPrepared)
    {
        Debug.Log("Preparing video");
        yield return waitTime;
        break;
    }
    Debug.Log("Done preparing video");
    // Passing the Texture from video to RawImage to be displayed
    displayer.texture = vidPlayer.texture;
    // Play Video
    vidPlayer.Play();
    // Play Sound
    audSource.Play();
    Debug.Log("Playing video");
    while (vidPlayer.isPlaying)
    {
        yield return null;
    }
    Debug.Log("Done playing video");
}

```

Potongan Kode 5-17 *videoPlayer.cs* Bagian 2: *playVid()*



Gambar 5-26 Tampilan *RawImage* sebagai *Video Player*

5.2.5.5 Pembuatan Animasi Mesin

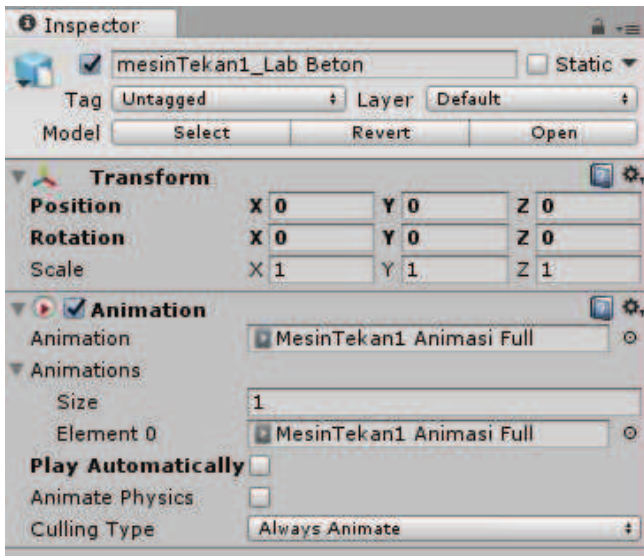
Dalam penelitian ini akan terdapat beberapa simulasi dari mesin-mesin yang terdapat di dalam laboratorium. Pembuatan animasi dari mesin akan dibuat menggunakan *Blender*. Dalam *Blender*, pembuatan animasi mesin adalah dengan membuat objek-nya terlebih dahulu. Setelah objek selesai dibuat, maka untuk membuat animasinya, dari tampilan *default view* blender, bisa dengan menekan *ctrl+left arrow key* sebanyak 2 kali untuk memasuki *layout* animasi. Untuk pembuatan animasinya sendiri bisa dengan pada *frame* yang diinginkan, posisikan objek sedemikian rupa dan tekan ‘i’ dan pilih *location rotation*, sehingga data lokasi dan putaran dari objek yang dipindah tersebut disimpan untuk frame saat itu disimpan. Pastikan hal tersebut dilakukan ketika dalam *object mode*, bukan *edit mode*. Jika ingin membuat animasi-animasi dari blender secara terpisah dapat dilakukan dengan memuat sebagian dari animasi dalam *NLA Strips*.



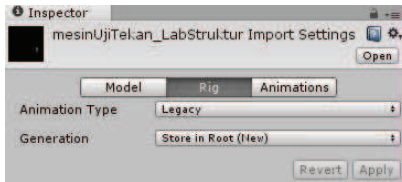
Gambar 5-27 Pembuatan Animasi dalam *Blender*

5.2.5.6 Pembuatan Interaksi dengan Mesin

Setelah animasi dibuat menggunakan *Blender*, maka objek tersebut dapat langsung ditaruh dalam folder *assets* proyek *Unity* sebagai *blend file* atau dengan diekspor terlebih dahulu sebagai *fbx file*. Selanjutnya objek tersebut dapat dimasukkan pada *scene Unity* dengan melakukan *drag-and-drop*, dan tidak lupa untuk menghilangkan tanda centang dari '*Play Automatically*' yang ditunjukkan pada Gambar 5-28 supaya objek tidak langsung menjalankan animasi tanpa perintah. Jadikan juga animasi dari objek tersebut sebagai *Legacy* dengan mengklik prefab objek tersebut, pada *tab Rig*, sebagaimana ditunjukkan pada Gambar 5-29.

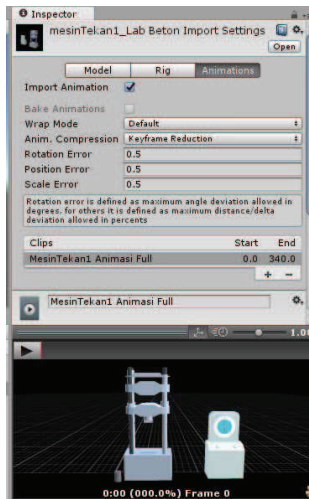


Gambar 5-28 Komponen Objek *blend* dengan Animasi saat Ditaruh Dalam *Scene*



Gambar 5-29 Ganti *Animation Type*

Untuk dapat menjalankan animasi, perlu dilihat dulu nama animasi yang dimiliki oleh objek tersebut. Animasi tersebut dapat dilihat apabila objek tersebut di-klik dan dilihat pada *inspector*, tab *Animation*. Terdapat bagian *Clips*. Nama tersebutlah yang jika terdapat lebih dari satu dapat dipanggil satu-persatu. Hal ini bisa digunakan ketika animasi ada lebih dari satu ketika objek dibuat menggunakan *NLA Strips*.



Gambar 5-30 *Inspector* dari Objek *blend*

Untuk dapat menjalankan animasi tersebut tanpa dalam *scene* tanpa menggunakan *Play Automatically*, adalah dengan menggunakan *script*. Animasi sebelumnya dapat dilihat pada Gambar 5-30 bahwa sudah ada animasi dari objek tersebut.

Untuk memanggil animasi tersebut, tahap pertama adalah diawali dengan mengambil objek tersebut menggunakan Potongan Kode 5-18, yang lalu dilanjutkan dengan mengambil komponen animasi dari objek tersebut dengan menggunakan kode seperti Potongan Kode 5-19, dan lalu dapat dipanggil pada *Update()* dengan menggunakan fungsi seperti apa yang ditunjukkan pada Potongan Kode 5-20. Apabila animasi tidak tergabung menjadi 1, animasi dapat dipanggil menggunakan Potongan Kode 5-21.

```
public GameObject mesinTekan1;
private Animation animesinTekan;
```

Potongan Kode 5-18 Variabel Objek dan Animasi

```
animesinTekan = mesinTekan1.GetComponent<Animation>();
animesinTekan.GetComponent<Animation>().wrapMode = WrapMode.Once;
```

Potongan Kode 5-19 Pengaturan Isi Variabel untuk Animasi

```
IEnumerator animasiMesinTekan1()
{
    animesinTekan.Play();
    return null;
}
```

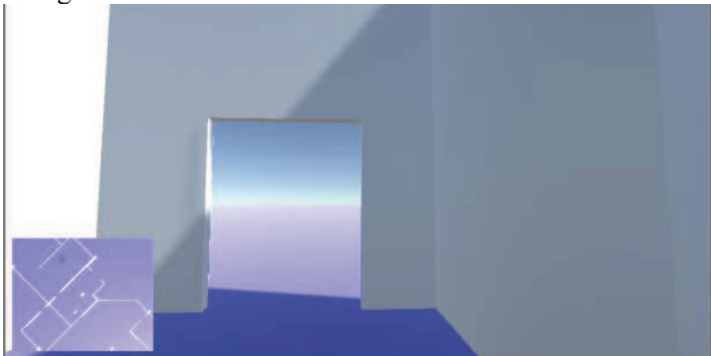
Potongan Kode 5-20 Menjalankan Animasi *Default*

```
IEnumerator animasiMesinStruktur()
{
    animesin.Play("Objek_MesinTekan_bisaBengkok.001Action");
    animesin["BasePenekanAction"].layer=2;
    animesin.Play("BasePenekanAction");
    animesin["Meteran_JarumAbu_LStrukturAction"].layer=1;
    animesin.Play("Meteran_JarumAbu_LStrukturAction");
    animesin.Play("Meteran_JarumMerah_LStrukturAction");
    return null;
}
```

Potongan Kode 5-21 Menjalankan Animasi Berdasarkan Potongan Animasi

5.2.5.7 Pembuatan *Mini-Map*

Untuk membantu dalam melakukan navigasi akan dibuatkan *mini-map* yang berperan untuk menunjukkan lokasi sekitar *FPS Controller* dari tampak atas seperti ditunjukkan pada Gambar 5-31. Untuk membuat *mini-map*, dapat dilakukan dengan menaruh satu kamera, diposisikan menghadap bawah, dan untuk *value* dari komponen *Transform* dan *Camera* yang terdapat dalam objek kamera *mini-map* tersebut dapat diberikan *value* sebagaimana ditunjukkan pada Gambar 5-32. Untuk Komponen *script* supaya kamera tersebut mengikuti *FPS Controller* dapat dengan diberikan *script* yang ditunjukkan pada Potongan Kode 5-22.



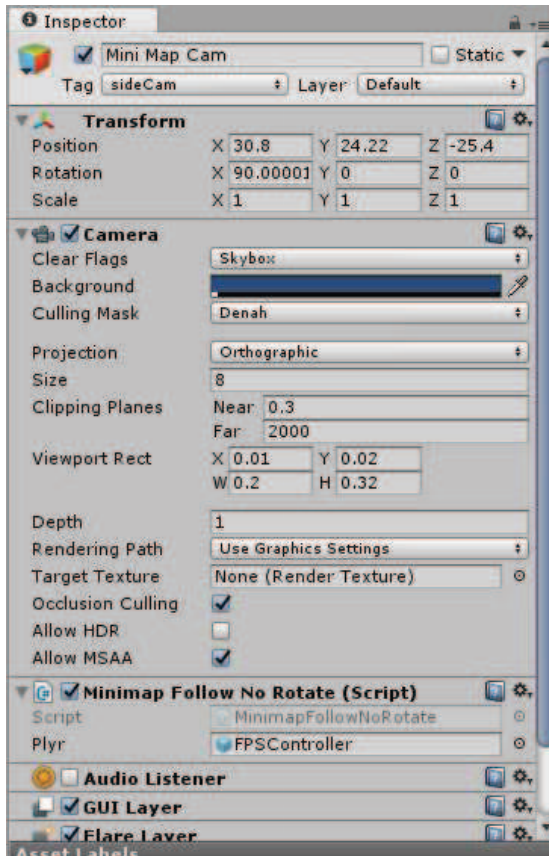
Gambar 5-31 Tampilan *Mini-Map*

```

1  using UnityEngine;
2  using System.Collections;
3
4  public class MinimapFollowNoRotate : MonoBehaviour {
5      public GameObject plyr;
6
7      public void Update ()
8      {
9          transform.position = new Vector3(plyr.transform.position.x, transform.position.y, plyr.transform.position.z);
10     }
11 }
12

```

Potongan Kode 5-22 *Script* untuk Mengikuti Player



Gambar 5-32 Komponen dari Kamera *Mini-Map*

5.2.6 Pencahayaan

Dalam aplikasi peta interaktif ini, tetap diperlukan elemen yang dapat memberikan terang tidaknya suatu lokasi. Untuk bagian antarmuka pengguna tidak memerlukan pencahayaan, namun untuk *GameObject* yang terdapat di dalam Unity diperlukan cahaya. Di dalam Unity Editor terdapat fitur untuk menambahkan cahaya seperti cahaya matahari dan lampu. Seberapa terang tidaknya cahaya tersebut juga dapat diatur dalam Unity. Program Blender yang digunakan dalam

penelitian ini untuk membuat model gedung dan mesin juga memiliki fitur untuk memberikan pencahayaan, namun pencahayaan yang dibuat melalui Blender tidak dapat di-eskpor ke dalam Unity.

5.2.7 Pengaturan Akhir

Setelah semua *GameObject* dan *Script* telah dibuat dan disatukan ke dalam satu atau beberapa *scene*, maka dapat diatur untuk bagaimana *buid* akan dilakukan pada peta interaktif ini. Pengaturan ini dapat diatur melalui *project setting* dan *build setting*. Hal ini juga tidak ada salahnya dilakukan pada saat awal perancangan peta interaktif ini, dikarenakan peta interaktif ini akan ditargetkan untuk dapat dimainkan pada web browser, sedangkan ada beberapa fitur *scripting* yang tidak bisa dijalankan pada *web browser*. Untuk *build* yang lebih optimal, pastikan tidak ada aset yang tidak terpakai, dan awal pembuatan *Unity Project* ini tidak menggunakan *Unity Editor* versi lama, karena terkadang akan timbul masalah ketika melakukan *build* mulai dari waktu *build* yang lama, tidak bisa dilakukan *build*, atau *build* berhasil dijalankan namun tidak dapat dijalankan. Untuk melakukan *build*, pilihannya terdapat pada menu *file*, tentukan sistem yang akan dituju sebagai sarana untuk menjalankan aplikasi, diman pada penelitian ini adalah menggunakan WebGL, tentukan lokasi keluarannya, dan menunggu hasil dari *build*. Pastikan *build* terbuat dalam sebuah *folder* sendiri, sehingga tidak tercampur dengan file lain. Untuk dapat menjalankan hasil *build* tersebut, taruh *folder* yang menyimpan aplikasi tersebut dalam *host*, lalu buka folder tersebut atau *html file* yang ada dalam *folder* tersebut.

5.3 Uji Coba Sistem

Dalam proses uji coba sistem akan dibagi menjadi dua bagian, yakni uji coba kebutuhan fungsional dan uji coba kebutuhan non-fungsional, dimana untuk menguji kebutuhan fungsional seluruhnya akan terkait dengan *test case*, dan untuk kebutuhan non-fungsional hanya sebagian yang terkait dengan *test case* dan sisanya akan diuji sendiri tanpa *test case* untuk

mempermudah perbandingan. Untuk daftar singkat dari *test case* dapat dilihat pada Tabel 5-3, dimana ditunjukkan juga apabila *test case* tersebut meliputi kebutuhan fungsional, maka akan terdapat ID KF dan apabila meliputi kebutuhan non-fungsional akan terdapat ID KNF. Untuk data lengkap *test case* dapat dilihat pada Lampiran D.

Tabel 5-3 Daftar *Test Case* beserta *Use Case* dan Kebutuhan Fungsional maupun Non-Fungsional Terkait

TC ID	Nama Test Case	ID UC/KF/KNF
TC-01	Membuka aplikasi saat aplikasi di dalam <i>host</i>	UC-101, KF-01
TC-02	Membuka aplikasi tanpa ditaruh dalam <i>host</i>	UC-101, KF-01
TC-03	Menampilkan menu <i>about</i> dengan koneksi <i>host</i>	UC-102, KF-02
TC-04	Menampilkan menu <i>about</i> tanpa koneksi <i>host</i>	UC-103, KF-03, KNF-04
TC-05	Pengguna menjelajahi peta interaktif	UC-103, KF-03
TC-06	Pengguna menjelajahi peta interkatif tanpa koneksi <i>host</i>	UC-014, KF-05, KF-06, KF-07, KNF-04
TC-07	Menjalankan simulasi mesin	UC-104, KF-05, KF-6, KF-07
TC-08	Menjalankan simulasi mesin tanpa koneksi <i>host</i>	UC-104, KF-05, KF-6, KF-07, KNF-04
TC-09	Uji tampilkan daftar video berdasarkan <i>database</i>	UC-105, UC-106, KF-08, KF-11, KNF-07
TC-10	Uji tampilkan daftar video tanpa koneksi <i>host</i>	UC-105, UC-106, KF-08, KF-11, KNF-04
TC-11	Uji menampilkan video	UC-107, UC-108, KF-08, KF-11, KNF-07
TC-12	Uji <i>pause menu</i>	UC-109, KF-09

TC ID	Nama Test Case	ID UC/KF/KNF
TC-13	Uji <i>mini-map</i>	UC-110, KF-10
TC-14	Uji unggah video dengan berbagai ukuran	UC-201, UC202, UC-203, KF-11, KNF-08
TC-15	Uji membuka aplikasi secara bersamaan	KNF-05
TC-16	Uji membuka video secara bersamaan	KNF-05

5.3.1 Uji Coba Fungsional

Kebutuhan-kebutuhan fungsional yang sudah tertera pada Tabel 4-1 sudah dibuatkan *Use Case*-nya berdasarkan kebutuhan tersebut yang ditampilkan pada Tabel 4-5. Hasil pengujian untuk

akan didapatkan dengan menjalankan aksi dari *test case* dan melihat perbandingan hasil yang diharapkan dengan hasil aktual. Berdasarkan hasil dari perbandingan tersebut maka akan didapatkan suatu kesimpulan apakah kebutuhan tersebut sudah cukup terpenuhi.

5.3.2 Uji Coba Non-Fungsional

Beberapa kebutuhan non-fungsional sudah diliputkan di dalam *test case* seperti apa yang ditunjukkan pada tabel Tabel 5-3. Dari hasil uji *test case* yang terkait, maka akan diambil kesimpulan apakah kebutuhan non-fungsional tersebut sudah tercukupi. Untuk kebutuhan non-fungsional sisanya akan dilakukan uji coba terpisah dari *test case* untuk mempermudah perbandingan dan karena beberapa tidak memiliki standar tersendiri.

5.3.2.1 Uji Kemampuan Aplikasi Saat Koneksi dengan Host Terputus

Pengujian akan dilakukan sesuai dengan *test case*, dimana bagian interaksi manakah yang akan tetap dapat berjalan apabila ketika aplikasi sudah dijalankan, koneksi dengan *host* telah terputus.

5.3.2.2 Uji Kemampuan Aplikasi Saat Diakses Bersamaan oleh Lebih dari Satu Pengguna

Pengujian ini akan dilakukn sesuai dengan *test case*. Aplikasi yang sudah ditaruh dalam *host* akan dibuka oleh lebih dari satu perangkat komputer, dan akan dilihat apakah aplikasi akan tetap dapat berjalan tanpa ada masalah.

5.3.2.3 Uji Kemampuan Aplikasi Ketika Ada Perubahan dalam Database

Pengujian akan mengikuti *test case* yang meliputi pengambilan data dari *database*. Apabila aplikasi dapat menampilkan sesuai dengan *database* maka bagian ini dapat dikatakan suksse

5.3.2.4 Uji Kemampuan Aplikasi untuk Menerima Video dengan Ukuran Besar

Pengujian akan dilakukan sesuai dengan *test case*, dimana akan dicoba untuk melakukan pengunggahan video terlebih dahulu menggunakan aplikasi web dan memeriksa batas ukuran *file* maksimal yang dapat diterima melalui aplikasi web dan apakah terdapat batas ukuran *file* yang dapat diterima oleh aplikasi *Unity*.

5.3.2.5 Uji Performa, Uji Platform Web, Uji Kompatibilitas Browser, dan Uji Kompatibilitas Sistem Operasi

Tiga kebutuhan non-fungsional dimana akan dilakukan pengujian jumlah *frame* per detiknya untuk menguji performa, lama load dari aplikasi untuk menguji *platform web*, dan dicoba pada beragam *browser* untuk menguji apakah aplikasi dapat berjalan pada *browser* tertentu, dimana ketiga kebutuhan non-fungsional ini dipisahkan dari *test case* karena tidak memiliki standar dan untuk mempermudah perbandingan.

5.3.2.6 Evaluasi Implementasi 3D

Hal lain yang dapat diuji adalah apakah model-model dari gedung beserta isi dari gedung Jurusan Teknik Sipil ITS sudah sesuai dengan aslinya. Evaluasi akan dilakukan dengan memberikan perbandingan antara foto aslinya dengan model tiga dimensi yang telah dibuat.

Halaman ini sengaja dikosongkan

BAB VI

HASIL PEMBAHASAN UJI COBA

Pada bab ini akan dijelaskan hasil uji coba mengikuti hasil dari uji coba yang dijelaskan pada bab sebelumnya. Hasil dari uji coba ini akan ditampilkan apa adanya.

6.1 Hasil Uji Coba Fungsionalitas

Hasil dari uji coba fungsionalitas merupakan langkah yang dilakukan untuk menguji aplikasi dengan menggunakan unit test dari rancangan test case yang dapat dilihat pada Lampiran D. Setiap skenario pada test case dapat dilihat pada Tabel 6-1.

Tabel 6-1 Hasil dari *test case*

ID Test Case	Hasil
TC-01	Sukses
TC-02	Gagal
TC-03	Sukses
TC-04	Gagal
TC-05	Sukses
TC-06	Sukses
TC-07	Sukses
TC-08	Sukses
TC-09	Sukses
TC-10	Gagal
TC-11	Sukses
TC-12	Sukses
TC-13	Sukses
TC-14	Gagal
TC-15	Sukses
TC-16	Sukses

Berdasarkan hasil dari *test case* yang tercantum pada Tabel 6-1 maupun Lampiran D, dapat dikatakan kebutuhan fungsional dan *use case* semuanya terpenuhi karena dapat dijalankan di dalam aplikasi. Untuk bagian *test case* yang dinyatakan gagal adalah bagian dimana pengujian kebutuhan non-fungsional disangkutpautkan.

6.2 Hasil Uji Coba Non-Fungsional

Melihat hasil dari *test case* yang terlampir pada Tabel 6-1 ataupun Lampiran D, dapat dilihat adanya kegagalan pada beberapa *test case* yang dikaitkan dengan kebutuhan fungsional, seperti kemampuan aplikasi ketika koneksi dengan *host* terputus, dan sebagian dari kebutuhan non-fungsional yang akan dilakukan uji sendiri di luar *test case* yang telah dibuat.

6.2.1 Hasil Uji Kemampuan Aplikasi saat Koneksi dengan Host Terputus

TC-04, TC-06, TC-08, dan TC-10 yang diterakan pada Tabel 6-1 menunjukkan hasil dari kebutuhan non-fungsional mengenai akses aplikasi ketika koneksi dengan *host* terputus. Interaksi smengelilingi peta dan menjalankan simulasi mesin tidak akan mengalami masalah apabila ketika sudah di-load, karena data-data tersebut sudah disimpan dalam *browser* dimana diperkirakan sebagai *cookies* atau *cache*. Namun untuk interaksi yang membutuhkan adanya koneksi dengan *host* tidak dapat berjalan karena tidak bisa mendapatkan data. Untuk bagian ini dalam uji coba tetap berjalan menampilkan panel antarmuka, namun tidak menunjukkan adanya isi teks, karena isi teks seharusnya didapatkan dengan mengambil dari *host*.

6.2.2 Hasil Uji Kemampuan Aplikasi saat Diakses oleh Lebih dari Satu Pengguna

Pengujian ini dikaitkan pada TC-15 dan TC-16. Ketika aplikasi dijalankan secara bersamaan, tidak ada masalah yang terjadi dari aplikasi tersebut. Pada awalnya diperkirakan adanya kemungkinan masalah muncul pada saat uji memainkan video, namun ternyata tidak ada masalah sama sekali. Hal ini

diperkirakan karena ukuran *file* video yang kecil dan koneksi masih cepat karena pada saat pengujiannya menggunakan *server* local dan *server* dalam laboratorium IKTI Jurusan Sistem Informasi ITS yang dapat dikatakan cukup cepat juga.

6.2.3 Hasil Uji Kemampuan Aplikasi Ketika Ada Perubahan dalam *Database*

Hasil pengujian kemampuan aplikasi akan hal ini dapat terlihat pada TC-09 dan TC-11, karena pengujian dari bagian itu mengambil data dari *database* dan isinya akan ikut berubah berdasarkan *database*, sehingga diperlukan pula koneksi dengan *host*, yang mana apabila tidak ada maka tidak akan ada data baru yang masuk.

6.2.4 Hasil Uji Kemampuan Aplikasi untuk Menerima Video dengan Ukuran Besar

Pengujian ini dicakup pada TC-14, dimana awalnya dalam aplikasi web akan dicoba mulai dari 9MB, 19MB, 29MB, hingga 39MB. Hal ini untuk memeriksa batas maksimal *file* yang dapat diunggah dalam *database*. Saat dilakukan percobaan, diketahui *file* berhasil diunggah, namun tidak berhasil ketika ukuran dari *file* tersebut lebih besar daripada 20MB. Sehingga *file* yang dapat diunggah dalam *database* dapat dikatakan hanya *file* berukuran kecil. Untuk penambahan jumlah maksimal ukuran *file* yang dapat diunggah akan diperlukan pengaturan *server*. *File* yang kecil tersebut sepertinya juga mempengaruhi hasil uji dari TC-16, karena *file* dengan ukuran lebih besar dari 20MB tidak sempat diujikan.

6.2.5 Hasil Uji Performa, Uji *Platform Web*, Uji Kompatibilitas *Browser*, dan Uji Kompatibilitas Sistem Operasi

Dalam pengujian bagian ini akan dilampirkan terlebih dahulu spesifikasi perangkat yang akan berperan untuk menjalankan prototipe aplikasi yang telah dibuat. Spesifikasi perangkat uji coba ditunjukkan pada Tabel 6-2.

Tabel 6-2 Spesifikasi Perangkat Pengujian

Spesifikasi Perangkat Pengujian 1	
CPU	Intel® Core™ i3-3110M CPU @ 2.40GHz (4 CPUs) ~2.4GHz
RAM	4096MB RAM
GPU	Inte® HD Graphics 4000; Memory 1792MB
OS	Windows 8.1 Single Language 64-Bit (6.3, Build 9600)
Spesifikasi Perangkat Pengujian 2	
CPU	Intel® Core™ 2 Duo CPU E7500 @ 2.93GHz (2 CPUs), ~2.9GHz
RAM	4096MB RAM
GPU	AMD Radeon HD 6570 2805 MB
OS	Windows 7 Professional 64-bit (6.1, Build 7601)
Spesifikasi Perangkat Pengujian 3	
CPU	Intel Core i7 2 GHz
RAM	8 GB 1600 MHz DDR3
GPU	Intel Iris Pro 1536 MB
OS	MacOS Sierra Version 10.12.1
Spesifikasi Perangkat Pengujian 4	
CPU	Intel® Core™ i3 CPU M370 @ 2.40GHz (4 CPUs), ~2.4GHz
RAM	2048MB
GPU	ATI Radeon HD 5450 512MB
OS	Ubuntu 10.04 LTS
Spesifikasi Perangkat Pengujian 5	
CPU	Intel® Core™ i5-6400 CPU @ 2.70GHz (4 CPUs), ~2.7GHz
RAM	8192MB RAM
GPU	Intell® HD Graphics 530, Display Memory 128 MB, Shared Memory 3945MB
OS	Windows 10 Pro 64-bit (10.0 Build 15063)
Spesifikasi Perangkat Pengujian 6	
CPU	Intel® Core™ i3-2310M CPU @ 2.10GHz (4 CPUs), ~2.1GHz
RAM	4096 MB
GPU	Intel® HD Graphics 3000, VRAM 32MB, shared memory 1632 MB
OS	Windows 10 Pro 64-bit (10.0 Build 15063)

Setelah didapatkan spesifikasi perangkat tersebut, maka akan dilanjutkan dengan terlebih dahulu mencoba menjalankan aplikasi pada masing-masing perangkat dengan menggunakan *Mozilla Firefox* untuk dicatat waktu load dan *frame per second (FPS)* ketika dilakukan navigasi dan animasi. *Frame per second* dapat dimunculkan dengan menggunakan *utility FPS Counter* yang disediakan dalam *Unity Standard Assets*. Hasil dari uji coba tersebut dapat dilihat pada Tabel 6-3

Tabel 6-3 Loadtime dan Rerata Frame per Second

Sistem	Waktu Load	Rerata FPS
1	9 detik	40-60 FPS ketika berkeliling; 20-40 FPS ketika <i>play</i> video;
2	7 detik	40-60 FPS ketika berkeliling; 30-40 FPS ketika <i>play</i> video;
3	8 detik	40-60 FPS ketika berkeliling; 20-30 FPS ketika <i>play</i> video;
4	11 detik	40-50 FPS ketika berkeliling; 20-30 FPS ketika <i>play</i> video;
5	10 detik	50-60 FPS ketika berkeliling; 20-30 FPS ketika <i>play</i> video;
6	20 detik	20-28 FPS ketika berkeliling; 10-25 FPS ketika <i>play</i> video

Berdasarkan hasil uji coba pada Tabel 6-3, dapat terlihat semua aplikasi dapat berjalan pada *Windows*, *MacOS*, dan *Linux*, sehingga dapat dikatakan tidak adanya masalah mengenai kompatibilitas sistem operasi. Untuk waktu *load* sepertinya spesifikasi perangkat akan berpengaruh, walaupun memiliki kemungkinan waktu *load* dapat berbeda-beda tergantung dari kecepatan akses *host* dan alokasi *memory* yang terpakai saat itu, karena pada perangkat yang sama, waktu *load* sendiri dapat membutuhkan waktu yang berbeda sekian detik.

Untuk rerata *frame* per detik pada dasarnya paling tinggi adalah ketika kamera dari karakter tidak menghadap terlalu banyak objek, karena ketika kamera tersebut menghadap objek yang

banyak maka akan perlu melakukan render terhadap objek-objek tersebut. Ketika dilakukan simulasi mesin, perubahan *frame* detik hanya turus sekian *frame*, tidak menimbulkan gangguan stabilitas *frame* per detik. Ketika video dalam aplikasi dimainkan barulah terjadi penurunan dalam *frame* per second. Dapat dilihat dari hasil pengujian, *frame* per second jatuh hamper setengahnya. Hal ini sepertinya dikarenakan dibutuhkan tenaga *rendering* lagi untuk munculnya video, sementara ada juga bagian dari peta 3D masih terlihat pada kamera, sehingga terjadi 2 *rendering*, sehingga menimbulkan jatuhnya *frame* per detik. Adapun kecurigaan lain adalah berdasarkan VGA yang dimiliki oleh perangkat, namun saat di-test pada perangkat 5 yang menggunakan *integrated VGA* dari *Intel*, aplikasi tetap dapat berjalan lancar. Diperkirakan adanya faktor lain yang membantu seperti processor dan RAM dalam menjalankan aplikasi. Adapun digunakan laptop dengan spesifikasi yang lebih rendah lagi seperti pada perangkat 6, barulah didapatkan rerata FPS yang lebih rendah saat berkelling, yakni tidak lebih besar dari 30 FPS. Untuk lama load variasi yang ada hanya hitungan sekian detik, atau ketika baru pertama kali dibuka pada perangkat tersebut maka dibutuhkan waktu sekitar 20 detik.

Tabel 6-4 Kompatibilitas Browser


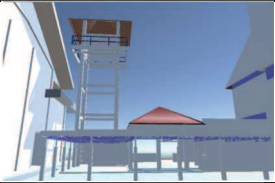



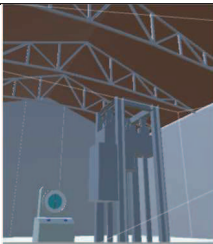
Browser	Hasil
<i>Mozilla Firefox (Ver. 54.0.1 32-bit)</i>	Berjalan
<i>Google Chrome (Ver. 59.0.3071.115 64-bit)</i>	Berjalan
<i>Internet Explorer 11 (Ver. 11.0.9600.18739)</i>	Tidak berjalan
<i>Opera (Ver. 46.0.2597.57)</i>	Berjalan


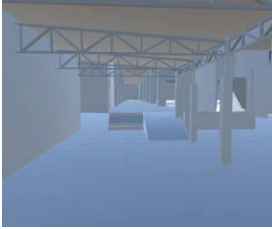


Berdasarkan Tabel 6-4 dapat terlihat bahwa aplikasi tetap dapat berjalan pada beberapa *web browser* mengecualikan *Internet Explorer*. Pada saat dijalankan sudah terdapat *warning* dari sistem yang mengatakan bahwa *browser* tidak kompatibel.

6.2.6 Hasil Evaluasi Implementasi 3D

Evaluasi mengenai implementasi 3D dilakukan dengan membandingkan hasil model 3D dengan foto nyata bangunan. Perbandingan dari keduanya dapat dilihat pada Tabel 6-5. Dapat dilihat bahwa dari bentuk gedung dan mesinnya apabila dibandingkan dengan foto yang sudah diambil sebelumnya, sekurang lebih-nya sudah mirip dengan aslinya, hanya saja masih perlu pengerjaan lebih dalam menggunakan tekstur dan material pada model supaya terlihat nyata.

Tabel 6-5 Implementasi 3D

Objek	Kondisi Nyata	Peta 3D
Tower dekat kantin		
Gedung A, B, dan C		
Workshop Struktur		

Objek	Kondisi Nyata	Peta 3D
Kantin		
Lab Beton		

BAB VII

KESIMPULAN DAN SARAN

7.1 Kesimpulan

Berdasarkan hasil dari pengerjaan tugas akhir ini, dapat diambil kesimpulan sebagai berikut:

1. Dengan adanya Unity5, terdapat fitur-fitur yang dapat dipakai oleh free user dimana yang sebelumnya tidak dapat dipakai oleh versi-versi sebelumnya yang hanya bisa digunakan oleh Unity premium
2. Fitur WWW dalam Unity paling mudah untuk mengambil String, Image, dan Video. Untuk gameobject bisa digunakan fitur *assetBundle* sebagai *compressed file* yang menyimpan data-data game object dan script apa saja yang akan menempel di dalamnya, namun assetbundles ini tidak dapat menyimpan script. Sehingga apabila file .unity3D yang saat ini tidak mengandung script baru di-attach pada assetbundles yang baru, maka script tidak akan berjalan.
3. Untuk mengunggah script diperlukan cara yang jauh lebih merepotkan daripada menggunakan Unity Editor. Sehingga jika ingin mengadakan script baru, akan lebih mudah jika melakukan build ulang. Untuk hal-hal seperti image, string, video, dan gameObject, hal-hal tersebut masih bisa diunggah ke database atau web hosting
4. Dalam mengunggah file video ke dalam database, saat dicoba pada laptop milik penulis, XAMPP sempat tidak mau menunjukkan list tabel yang berisikan video dalam bentuk enkripsi byte64, dan baru akan muncul apabila diambil melalui query.
5. Untuk animasi, menaruh .blend pada asset Unity akan lebih mudah dalam mengubah objek daripada .fbx

karena tidak perlu melakukan export lagi dan perbandingan ukuran meter antar Unity dengan Blender sudah kurang lebih sama, sehingga tidak perlu mengatur scaling dari model yang dibuat diluar Unity.

6. Untuk aplikasi *web* masih bisa dikembangkan lagi supaya lebih efisien.
7. Apabila aplikasi masih dirasa berat, dapat dilakukan perbedaan level antara gedung sehingga dapat meringankan *rendering* dan meningkatkan FPS

7.2 Saran

Saran untuk penelitian selanjutnya dalam hal pembuatan plikasi peta 3D berdasarkan penelitian ini adalah:

1. Untuk ke depannya, supaya pihak klien lebih mudah untuk mengganti konten yang simpel seperti gambar, tulisan, dan video, alangkah lebih baik jika menggunakan fitur WWW dalam Unity. Sehingga tinggal dibuatkan suatu aplikasi web berbasis PHP yang berfungsi untuk mengatur isi dari database tersebut.
2. Untuk upload script baru pada database sebenarnya bisa, namun hal tersebut hanya berlaku menggunakan C# dan tahap yang perlu dilakukan ada banyak sehingga merepotkan. Sehingga untuk Unity5 saat ini sebaiknya jika ingin menambahkan script baru, untuk melakukan build ulang.
3. Jika pengembangan aplikasi *web* ingin dikembangkan lebih dalam lagi, sebaiknya pada penelitian selanjutnya tidak perlu terlalu banyak membuat gedung dan animasi, cukup fokus pada koneksi *web* dengan *Unity*, karena masing-masing membutuhkan waktu yang lama karena butuhnya waktu untuk mempelajari masing-masing dan karena bangunan memiliki struktur yang berbeda.
4. Pada bagian memainkan video dalam aplikasi, apabila *load* video dirasa terlalu lama, bisa mencoba fitur *stream* yang ada dalam *Unity*

DAFTAR PUSTAKA

- [1] E. Overby, "Process Virtualization Theory," *Organization Science*, vol. vol. 19, no. no. 2, pp. pp. 277-291, 2008.
- [2] S. Wang, Z. Mao, C. Zeng, H. Gong, S. Li and B. Chen, "A New Method of Virtual Reality Based on Unity3D," in *2010 18th International Conference on Geoinformatics*, Beijing, China, 2010.
- [3] F. M. Purnama, Pengembangan Peta Interaktif Tiga Dimensi Jurusan Teknik Sipil Institut Teknologi Sepuluh Nopember Menggunakan Unity Engine, Surabaya: ITS, 2011.
- [4] P. S. Endragian, Pengembangan Peta Tiga Dimensi Interaktif Berbasis Web Menggunakan Unity (Studi Kasus: Perpustakaan Institut Teknologi Sepuluh Nopember Surabaya, Surabaya: ITS, 2015.
- [5] J. Xie, "Research on Key Technologies Base Unity3D Game Engine," in *2012 7th International Conference on Computer Science & Education (ICCSE)*, Melbourne, Australia, 2012.
- [6] A. Thorn, Game Engine Design and Implementation, Jones & Bartlett Learning, 2011.
- [7] T. Kaneko, Teknologi Perpetaan Digital. Kursus Singkat Dasar dan Aplikasi pPemetaan Digital, UGM: Tidak diterbitkan, 1995.
- [8] D. Rosenberg and M. Stephens, Use Case Driven Object Modeling with UML: Theory and Practice, New York, USA: Apress, 2013.

Halaman ini sengaja dikosongkan

BIODATA PENULIS



Penulis bernama lengkap I Gusti Bagus Rogeri Ariefianto, yang lahir di Surabaya pada tanggal 24 September 1994. Penulis merupakan anak kedua dari 3 bersaudara. Sebelumnya penulis menempuh Pendidikan di Sekolah Nasional Plus Jembatan Budaya, Kuta, Bali. Pengalaman keorganisasian yang pernah diallui oleh penulis ada beberapa yakni sebagai anggota PSDM Paduan Suara ITS dan anggota dalam negri Lembaga

Minat Bakat ITS. Selain bidang akademik tertarik juga untuk mendalami bidang non akademik seperti bermusik dan menggambar. Untuk mendapatkan gelar sarjana komputer di Jurusan SIstem Informasi ITS Surabaya, penulis mengambil laboratorium IKTI karena seseuai dengan minat penulis.

Halaman ini sengaja dikosongkan

LAMPIRAN A *USE CASE*

Use Case Descriptions

Tabel 10-1 *Use Case Description* untuk UC-101 Pengguna Umum
Membuka Menu Utama

UC-101: Pengguna umum membuka menu utama	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: <ul style="list-style-type: none">• Aplikasi berada dalam <i>host</i>• Saat aplikasi sedang di-<i>load</i>, koneksi dengan <i>host</i> tidak terputus• Aplikasi dibuka menggunakan <i>browser</i>.	
Triggers: Awal aplikasi dibuka dengan <i>browser</i>	
Basic Course: <ol style="list-style-type: none">1. Pengguna membuka <i>browser</i>2. Dalam <i>textfield</i> URL, pengguna memasukkan link lokasi URL aplikasi peta 3D yang berbentuk .html	
Post-Conditions: Menu utama dari aplikasi peta tiga dimensi keluar	
Alternate Courses: <ul style="list-style-type: none">• Pengguna membuka aplikasi dengan memasukkan link menuju folder yang mengandung file .html tersebut	

Tabel 10-2 Use Case Description Untuk Pengguna Umum Membuka Menu About

UC-102: Pembuka umum membuka menu <i>about</i>	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: Pengguna sedang di bagian menu utama	
Triggers: <i>Button</i> menu <i>about</i> di-klik	
Basic Course: Pengguna melihat adanya <i>button</i> dengan teks <i>about</i> dan meng-klik <i>button</i> tersebut untuk menampilkan antarmuka yang menjelaskan aplikasi	
Post-Conditions: UI menu <i>about</i> ditampilkan	
Alternate Courses:	

Tabel 10-3 Use Case Description untuk Pengguna Umum melakukan Navigasi dalam Peta 3D

UC-103: Pembuka Umum Melakukan Navigasi dalam Peta 3D	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: Pengguna umum melakukan klik pada <i>button</i> mulai pada menu utama.	
Triggers: Klik <i>button</i> mulai pada menu utama	
Basic Course: <ol style="list-style-type: none"> 1. Dari menu utama, pengguna melakukan klik pada <i>button</i> mulai di menu utama 2. Pengguna sudah bisa melakukan navigasi dalam peta tiga dimensi 	

3. Karakter yang digunakan pengguna adalah dalam bentuk pandangan orang pertama (*first person view*)
4. Karakter tersebut dapat digerakkan oleh pengguna
5. Karakter dapat bergerak maju, mundur, ke kiri, ataupun ke kanan, dengan menekan W, A, S, dan D *key*, atau dengan menekan *arrow keys*.
6. Karakter dapat melompat dengan ditekannya tombol spasi pada *keyboard*.
7. Karakter dapat bergerak lebih cepat atau berlari (*sprint*) dengan menekan tahan tombol *shift* pada *keyboard* sembari menekan tombol arah gerakan.
8. Untuk arah gerakan bisa ditekan secara bersamaan untuk digerakkan ke arah diantara arah tersebut selama tidak kedua arah yang ditekan tidak berlawanan
9. Karakter dapat melompat sambil bergerak dengan tombol arah
10. Karakter dapat melihat sekitar menyesuaikan dengan gerakan *mouse*.

Post-Conditions:

Pengguna sudah dapat melakukan navigasi pada peta 3D

Alternate Courses:

Tabel 10-4 *Use Case Description* Pengguna Umum Menjalankan Simulasi

UC-104: Pengguna Umum Menjalankan Simulasi	
<i>Primary Actor:</i> Pengguna Umum	<i>Level:</i> <i>User Goal</i>
<i>Pre-Condition:</i> Pengguna berada dalam <i>trigger area</i> untuk melakukan simulasi dari mesin terkait	
<i>Triggers:</i> Pengguna menekan tombol interaksi saat berada di dalam <i>trigger area</i> mesin terkait	
<i>Basic Course:</i> <ol style="list-style-type: none"> 1. Pengguna berjalan menuju <i>trigger area</i> dari mesin yang memiliki simulasi dan ingin dijalankan simulasinya 2. Pengguna menekan tombol interaksi 3. Simulasi dari mesin tersebut segera berjalan hingga selesai. 	
<i>Post-Conditions:</i> Mesin melakukan simulasi hingga selesai	
<i>Alternate Courses:</i>	

Tabel 10-5 Use Case Description Pengguna Umum Menampilkan daftar Video dari Laboratorium

UC-105: Pengguna umum menampilkan daftar video dari laboratorium	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: <ul style="list-style-type: none"> • Karakter pengguna sedang melakukan navigasi dalam peta tiga dimensi • Karakter pengguna sedang berada dalam trigger area untuk antarmuka video laboratorium 	
Triggers: Pengguna menekan tombol interaksi ketika karakter sedang berada dalam <i>trigger area</i> laboratorium terkait	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna menggerakkan karakter hingga <i>trigger area</i> laboratorium terkait 2. Pengguna menekan tombol interaksi ketika karakter masih berada dalam <i>trigger area</i> tersebut. 3. Tampilan antarmuka yang berisi daftar video dari laboratorium tersebut ditampilkan 4. Pengguna dapat menekan tombol <i>esc</i> pada keyboard untuk menutup antarmuka tersebut 	
Post-Conditions: Tampilan antarmuka berisi daftar video dari laboratorium terkait muncul	
Alternate Courses:	

Tabel 10-6 Use Case Description Pengguna Umum Mendapatkan Daftar Video Berdasarkan Database

UC-106: Pengguna umum mendapatkan daftar video berdasarkan <i>database</i>	
Primary Actor: Pengguna Umum	Level: User Goal
Pre-Condition: Berada dalam <i>trigger area</i> terkait dan tersambung dengan <i>host</i>	
Triggers: Tombol interaksi ditekan ketika karakter pengguna sedang berada dalam <i>trigger area</i> .	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna menggerakkan karkater hingga <i>trigger area</i> laboratorium terkait 2. Pengguna menekan tombol interaksi ketika karakter masih berada dalam <i>trigger area</i> tersebut. 3. Aplikasi mengambil data berdasarkan <i>link URL</i> mengenai daftar video lab terkait 4. Tampilan antarmuka yang berisi daftar video dari laboratorium tersebut ditampilkan 5. Pengguna dapat menekan tombol <i>esc</i> pada keyboard untuk menutup antarmuka tersebut 	
Post-Conditions: Menampilkan antarmuka daftar video dari laboratorium terkait berdasarkan <i>database</i>	
Alternate Courses:	

Tabel 10-7 *Use Case Description* Pengguna Umum Menampilkan Video

UC-107: Pengguna umum menampilkan video	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: Antarmuka daftar video sedang ditampilkan	
Triggers: Klik dilakukan pada salah satu <i>button</i> dari daftar video laboratorium terkait	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna menampilkan antarmuka daftar video laboratorium tertentu 2. Pengguna melakukan klik pada salah satu tombol dari daftar video tersebut 3. Pengguna berhasil memunculkan video tersebut dalam aplikasi. 4. Pengguna dapat menekan tombol <i>esc</i> pada <i>keyboard</i> untuk menutup video tersebut. 	
Post-Conditions: Ada video yang sedang dimainkan	
Alternate Courses:	

Tabel 10-8 Use Case Description Pengguna Umum Mendapatkan Video dari Database

UC-108: Pengguna umum mendapatkan video dari database	
Primary Actor: Pengguna Umum	Level: <i>User Goal</i>
Pre-Condition: Penggunaan umum sedang berada dalam <i>trigger area</i> laboratorium terkait	
Triggers: Tombol interaksi ditekan ketika karakter sedang berada dalam <i>trigger area</i> laboratorium terkait.	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna menampilkan antarmuka daftar video laboratorium tertentu 2. Pengguna melakukan klik pada salah satu tombol dari daftar video tersebut 3. Aplikasi mengambil link URL berdasarkan dari <i>button</i> mana yang di-klik, dimana URL tersebut berperan untuk mengambil video dari database. 4. Pengguna berhasil memunculkan video tersebut dalam aplikasi. 5. Pengguna dapat menekan tombol <i>esc</i> pada <i>keyboard</i> untuk menutup video tersebut. 	
Post-Conditions: Video dari <i>button</i> tersebut ditampilkan	
Alternate Courses:	

Tabel 10-9 Use Case Description Pengguna Umum Menampilkan Menu *Pause*

UC-109: Pengguna umum menampilkan menu <i>pause</i>	
Primary Actor: <i>Pengguna Umum</i>	Level: <i>User Goal</i>
Pre-Condition: <ul style="list-style-type: none"> • Pengguna sedang melakukan navigasi dalam peta tiga dimensi • Pengguna tidak sedang menampilkan video 	
Triggers: Tombol <i>esc</i> ditekan	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna sedang di dalam peta tiga dimensi dan dapat melakukan navigasi 2. Pengguna menekan tombol <i>esc</i> pada <i>keyboard</i> untuk melakukan <i>pause</i> 3. Antarmuka untuk menu <i>pause</i> muncul 4. Pengguna dapat menekan tombol <i>esc</i> pada <i>keyboard</i> atau menekan <i>button</i> dalam panel menu tersebut untuk kembali melakukan navigasi. 	
Post-Conditions: Menu <i>pause</i> muncul	
Alternate Courses:	

Tabel 10-10 Use Case Description Pengguna Umum Menampilkan *Mini-map*

UC-110: Pengguna umum menampilkan <i>mini-map</i>	
Primary Actor: Pengguna umum	Level: <i>User Goals</i>
Pre-Condition: Pengguna sudah dalam peta 3D dan sedang bisa melakukan navigasi	
Triggers: Memasuki peta 3D	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna memulai navigasi setelah dari menu utama 2. <i>Mini-map</i> secara otomatis akan ikut muncul pada layar 	
Post-Conditions: <i>Mini-map</i> muncul	
Alternate Courses:	

Tabel 10-11 Use Case Description Administrator mengakses halaman pengaturan web

UC-201: Administrator mengakses halaman pengaturan <i>web database</i>	
Primary Actor: Administrator	Level:
Pre-Condition: <ul style="list-style-type: none"> • Administrator membuka <i>browser</i> • Ada koneksi kepada <i>host</i> 	
Triggers: Membuka <i>link</i>	
Basic Course: Adminstrator membuka halaman <i>web</i> yang mengatur sebagian isi konten dari aplikasi	
Post-Conditions: Halman pengaturan isi konten ditampilkan	
Alternate Courses:	

Tabel 10-12 Use Case Description Administrator Melakukan Update atau Upload Teks pada Database

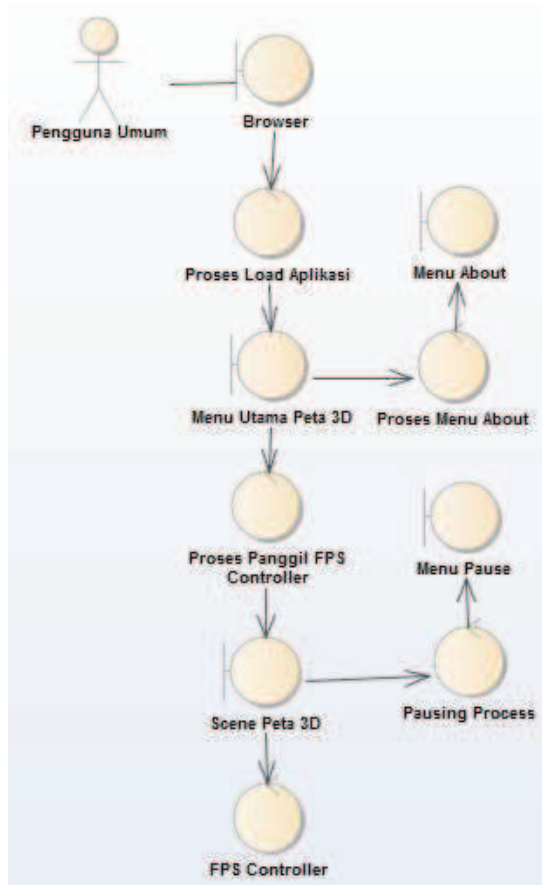
UC-202: Administrator melakukan update atau upload teks pada <i>database</i>	
Primary Actor: Administrator	Level:
Pre-Condition: Administrator terkoneksi dengan <i>host</i>	
Triggers: Unggah <i>file</i>	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna mengakses halaman web 2. Pengguna mengunggah video dengan judulnya 3. Judul dari video dilampirkan (jika mengunggah video baru) atau tergantikan (jika meng-<i>update</i> video lama) 4. Nama video yang akan ditampilkan dalam antarmuka daftar video akan berubah berdasarkan isi dari <i>database</i> yang terbaru 	
Post-Conditions: Video dan teks terunggah	
Alternate Courses: <ol style="list-style-type: none"> 1. Mengunggah file dengan ekstensi apapun 2. Mengunggah file dengan ukuran besar 3. Mengunggah file dengan nama yang panjang 	

Tabel 10-13 Use Case Description Administrator Melakukan Update atau Upload Video

UC-203: Administrator melakukan update atau upload video	
Primary Actor: Administrator	Level:
Pre-Condition: Halaman pengaturan <i>web</i> sedang diakses	
Triggers: Pengunggahan video	
Basic Course: <ol style="list-style-type: none"> 1. Pengguna mengakses halaman web 2. Pengguna mengunggah video dengan judulnya 3. Judul dari video dilampirkan (jika mengunggah video baru) atau tergantikan (jika meng-<i>update</i> video lama) 4. Video yang akan ditampilkan dalam antarmuka video akan berubah berdasarkan isi dari <i>database</i> yang terbaru 	
Post-Conditions: Video terunggah	
Alternate Courses:	

Halaman ini sengaja dikosongkan

LAMPIRAN B *ROBUSTNESS DIAGRAM*

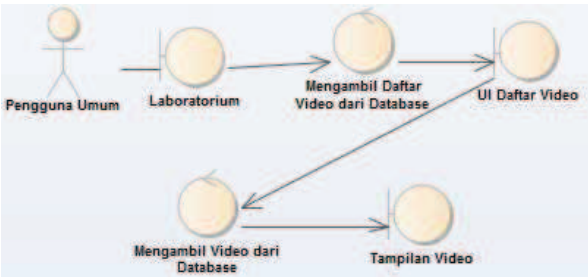


Gambar 11-1 Robustness Diagram untuk Menu About, Puase Menu, dan Navigasi

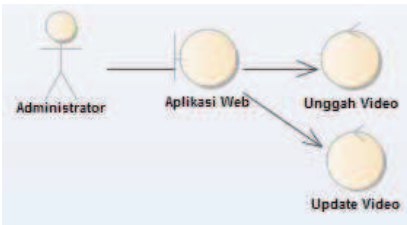
B-2



Gambar 11-2 *Robustness Diagram* untuk Pengguna Umum kepada Mesin dan Animasinya

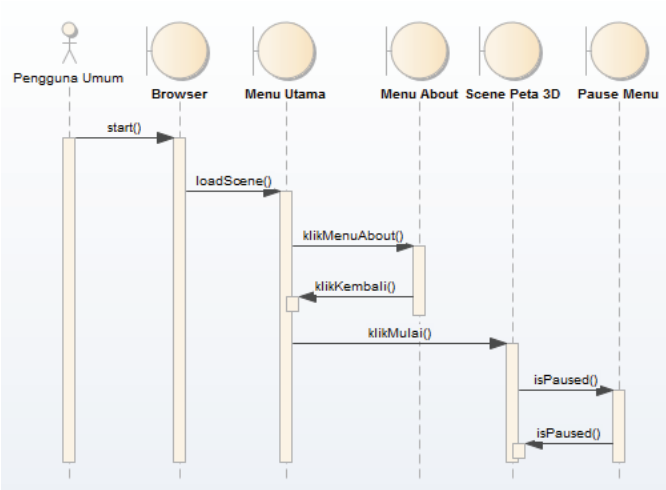


Gambar 11-3 *Robustness Diagram* untuk Pengguna Umum terhadap Daftar Video

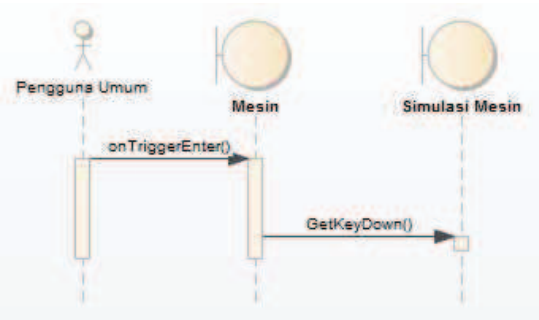


Gambar 11-4 *Robustness Diagram* untui Administrator dalam Mengatur Database

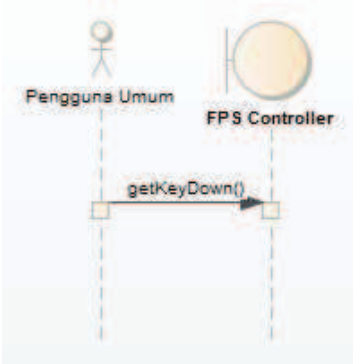
LAMPIRAN C *SEQUENCE DIAGRAM*



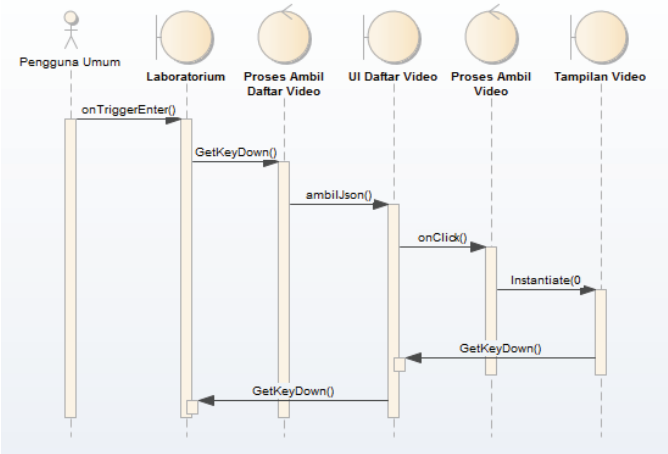
Gambar 12-1 *Sequence Diagram* untuk Membuka Menu Utama, Menu About, dan Pause Menu



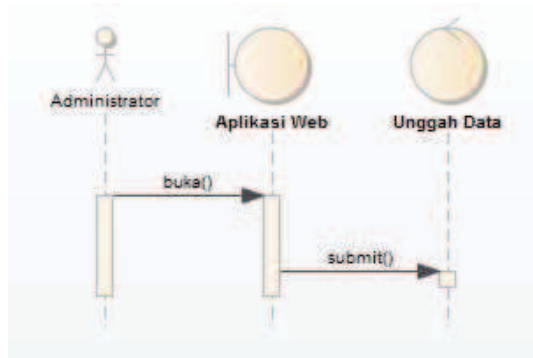
Gambar 12-2 *Sequence Diagram* untuk Menjalankan Simulasi Mesin



Gambar 12-3 *Sequence Diagram* untuk Navigasi Karakter



Gambar 12-4 *Sequence Diagram* untuk Menampilkan Daftar Video dan Memainkan Video



Gambar 12-5 *Sequence Diagram* untuk Administrator Melakukan Perubahan Data

Halaman ini sengaja dikosongkan

LAMPIRAN D TEST CASE

Tabel 13-1 TC-01: Membuka aplikasi dalam *host*

<i>Test Case ID</i>	TC-01			
<i>Nama Test Case</i>	Membuka aplikasi saat aplikasi ditaruh dalam <i>host</i>			
<i>Tujuan Test Case</i>	Menguji apakah aplikasi dapat berjalan ketika ditaruh di dalam <i>host</i>			
<i>Use Case Terkait</i>	UC-101			
<i>Kebutuhan Terkait</i>	KF-01			
<i>Pre-Condition</i>	Tersambung dengan <i>host</i> , aplikasi ditaruh dalam <i>host</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
1. Membuka browser 2. Menaruh link file .html aplikasi yang terletak di <i>host</i> pada <i>textfield</i> URL	Link URL file yang terletak di dalam <i>host</i>	Aplikasi dapat berjalan dan menunjukkkan menu utama	Aplikasi berjalan dan menunjukkan menu utama dari aplikasi	Sukses

Tabel 13-2 TC-02: Membuka aplikasi diluar *host*

Test Case ID	TC-02			
Nama Test Case	Membuka aplikasi tanpa ditaruh dalam <i>host</i>			
Tujuan Test Case	Menguji apakah aplikasi dapat dijalankan ketika aplikasi tersebut diletakkan dalam <i>local folder</i> .			
Use Case Terkait	UC-101			
Kebutuhan Terkait	KF-01			
Pre-Condition	Aplikasi berada di dalam folder local; Tersambung dengan <i>host</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
1. Membuka browser 2. Memasukkan lokasi <i>local file</i> .html aplikasi pada browser	Link <i>local file</i>	Aplikasi dapat berjalan dan menampilkan menu utama	Aplikasi tidak mau berjalan.	Gagal

Tabel 13-3 TC-03: Menampilkan menu *about* dengan koneksi *host*

Test Case ID	TC-03			
Nama Test Case	Menampilkan menu <i>about</i> dengan koneksi <i>host</i>			
Tujuan Test Case	Menguji apakah menu <i>about</i> dapat menampilkan teks dengan			
Use Case Terkait	UC-102			
Kebutuhan Terkait	KF-02			
Pre-Condition	Terdapat koneksi dengan <i>host</i> ; Aplikasi sudah terbuka pada <i>browser</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Menekan <i>button</i> about		Keluar teks yang terambil dari <i>database</i>	Teks sesuai dengan teks yang ditunjukkan oleh URL	Sukses

Tabel 13-4 TC-04: Menampilkan menu about tanpa koneksi *host*

Test Case ID	TC-04			
Nama Test Case	Menampilkan menu about tanpa koneksi <i>host</i>			
Tujuan Test Case	Menguji apakah menu <i>about</i> dapat menampilkan teks apabila tidak dapat tersambung dengan database dan menunjukkan <i>error</i>			
Use Case Terkait	UC-102			
Kebutuhan Terkait	KF-02, KNF-04			
Pre-Condition	Tidak ada koneksi dengan <i>host</i> ; Sedang dalam menu utama			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Menekan <i>button</i> about		Terjadi error	Tidak ada error. Panel dari menu <i>about</i> tetap muncul namun tidak ada teks apa-apa karena memang tidak ada teks yang terambil.	Gagal

Tabel 13-5 TC-05: Pengguna menjelajahi peta interaktif

Test Case ID	TC-05			
Nama Test Case	Pengguna menjelajahi peta interaktif			
Tujuan Test Case	Menguji apakah pengguna dapat melakukan navigasi di dalam peta 3D seperti berjalan dan melihat sekitar peta 3D			
Use Case Terkait	UC-103			
Kebutuhan Terkait	KF-03			
Pre-Condition	Sedang di menu utama; Terkoneksi dengan <i>host</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
1. Menekan <i>button</i> mulai dari main menu 2. Menekan tombol arah 3. Menekan spasi 4. Menekan spasi sambil berjalan		1. Muncul <i>First Person Controller</i> (Karakter yang dikontrol dalam game dengan sudut pandang orang pertama) 2. Karakter dapat bergerak sesuai arah	1. <i>First Person Controller</i> muncul 2. Karakter bergerak sesuai arah 3. Karakter melompat 4. Karakter melompat sesuai arah 5. Karakter bergerak lebih cepat	Sukses

5. Menahan tombol <i>shift</i> pada keyboard dan tombol arah untuk bergerak lebih cepat		3. Karakter dapat melompat	6. Kamera bergerak sesuai dengan gerakan <i>mouse</i>	
6. Menggerakkan <i>mouse</i>		4. Karakter dapat melompat sesuai arah berjalan		
		5. Karakter dapat bergerak lebih cepat		
		6. Kamera dapat melihat sekitar sesuai dengan arah gerakan mouse		

Tabel 13-6 TC-06: Pengguna menjelajahi peta interaktif tanpa koneksi *host*

Test Case ID	TC-06			
Nama Test Case	Pengguna menjelajahi peta interaktif tanpa koneksi ke <i>host</i>			
Tujuan Test Case	Menguji apakah karakter akan tetap dapat berkeliling peta tiga dimensi apabila tidak terdapat koneksi dengan <i>host</i> .			
Use Case Terkait	UC-103			
Kebutuhan Terkait	KF-03, KNF-04			
Pre-Condition	<i>FPS Controller</i> sudah terpanggil; Koneksi dengan <i>host</i> diputus			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Menekan tombol navigasi di dalam peta 3D		Karakter akan tetap dapat berkeliling	Karakter tetap bisa berkeliling	Sukses

Tabel 13-7 TC-07: Menjalankan simulasi mesin

Test Case ID	TC-07			
Nama Test Case	Menjalankan Simulasi Mesin			
Tujuan Test Case	Menguji apakah mesin atau objek tertentu di dalam aplikasi dapat menjalankan animasi			
Use Case Terkait	UC-104			
Kebutuhan Terkait	KF-05, KF-06, KF-07			
Pre-Condition	FPS Controller sudah terpanggil; Terdapat koneksi dengan host; Player berada dalam trigger area mesin terkait			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Pengguna menekan tombol interaksi saat karakter masih berada dalam trigger area tersebut.		Mesin dapat menjalankan animasi	Animasi dari mesin berjalan	sukses

Tabel 13-8 TC-08: Menjalankan simulasi mesin tanpa koneksi *host*

Test Case ID	TC-08			
Nama Test Case	Menjalankan simulasi mesin tanpa koneksi dengan <i>host</i>			
Tujuan Test Case	Menguji apakah simulasi di dalam aplikasi akna tetap dapat berjalan apabila koneksi dengan <i>host</i> telah terputus			
Use Case Terkait	UC-104			
Kebutuhan Terkait	KF-05, KF-06, KF-07, KNF-04			
Pre-Condition	<i>FirstPersonController</i> sudah terpanggil; Tidak ada koneksi dengan <i>host</i> ; <i>Player</i> berada dalam <i>trigger area</i> mesin			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Tombol interaksi ditekan ketika karakter masih berada dalam <i>trigger area</i> .		Animasi dari mesin tersebut akan tetap berjalan	Animasi mesin tetap dapat berjalan	Sukses

Tabel 13-9 TC-09: Uji Tampilkan Daftar Video berdasarkan *Database*

Test Case ID	TC-09			
Nama Test Case	Uji Tampilkan Daftar Video berdasarkan <i>database</i> .			
Tujuan Test Case	Menguji apakah daftar video akan muncul berdasarkan isi dari <i>database</i> terkait			
Use Case Terkait	UC-105, UC-106			
Kebutuhan Terkait	KF-08, KF-11, KNF-07			
Pre-Condition	<i>FirstPersonController</i> sudah terpanggil; Terdapat koneksi dengan <i>host</i> ; <i>Player</i> berada dalam <i>trigger area</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
1. Tombol interaksi ditekan selama berada dalam <i>trigger area</i> . 2. Ketika muncul antarmuka daftar video, tutup antarmuka tersebut	Pergantian value nama_video dalam database	1. Antarmuka daftar video memunculkan daftar video berdasarkan <i>database</i> 2. Ketika nama dalam databas diganti,	1. Daftar video sesuai dengan <i>database</i> 2. Daftar video dalam aplikasi tetap sesuai dengan <i>database</i>	Sukses

3. Lakukan perubahan nama judul video yang muncul pada daftar video tersebut.		maka ketika dijalankna ulang tahap 1 dan 2, daftar video dalam aplikasi juga akan ikut berubah		
4. Ulangi tahap 1 dan 2				

Tabel 13-10 Uji memanggil daftar video tanpa koneksi *host*

Test Case ID	TC-10			
Nama Test Case	Uji tampilkan daftar video tanpa koneksi <i>host</i>			
Tujuan Test Case	Menguji apakah daftar video akan tetap tampil tanpa koneksi ke <i>host</i>			
Use Case Terkait	UC-105, UC-106			
Kebutuhan Terkait	KNF-04, KF-08, KF-11			
Pre-Condition	<i>FirstPersonController</i> sedang dalam <i>trigger area</i> daftar video untuk lab terkait;			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Tekan tombol interaksi		<i>Error</i> muncul karena tidak ada koneksi dengan <i>host</i>	Tidak ada error. Panel dari daftar video tetap muncul tanpa isi	Gagal

Tabel 13-11 TC-11: Uji menampilkan video

Test Case ID	TC-11			
Nama Test Case	Uji menampilkan video			
Tujuan Test Case	Menguji apakah video dari <i>database</i> dapat dimainkan di dalam aplikasi			
Use Case Terkait	UC-107, UC-108			
Kebutuhan Terkait	KF-08, KF-11, KNF-07			
Pre-Condition	Daftar video sedang ditampilkan; Terhubung dengan <i>host</i>			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
1. Melakukan klik pada <i>button</i> dalam daftar video. 2. Tekan tombol <i>esc</i> untuk menutup video yang ditampilkan		Video dapat dimainkan dan sesuai dengan apa yang telah diunggah ke dalam database. Video dapat ditutup dengan menggunakan tombol <i>esc</i> .	Video berhasil dimunculkan dan sesuai dengan apa yang telah diunggah sebelumnya. Video bisa ditutup dengan tombol <i>esc</i>	Sukses

3. <i>Update</i> video dalam <i>database</i>				
4. Mainkan video tersebut.				

Tabel 13-12 TC-12: Uji *pause menu* saat melakukan navigasi

Test Case ID	TC-12			
Nama Test Case	Uji <i>pause menu</i> saat melakukan navigasi			
Tujuan Test Case	Menguji apakah permainan dapat dihentikan sementara			
Use Case Terkait	UC-109			
Kebutuhan Terkait	KF-09			
Pre-Condition	<i>FPS Controller</i> sudah terpanggil;			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Menekan tombol <i>esc</i>		Menu <i>pause</i> dan <i>mouse cursor</i> muncul, karakter tidak bisa bergerak.		

Tabel 13-13 TC13: Uji *mini-map*

Test Case ID	TC-13			
Nama Test Case	Uji <i>Mini-map</i>			
Tujuan Test Case	Menguji apakah <i>mini-map</i> dapat muncul dalam aplikasi			
Use Case Terkait	UC-110			
Kebutuhan Terkait	KF-10			
Pre-Condition	Sedang dalam menu utama;			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Dari menu utama, klik <i>button</i> mulai		Saat karakter terpanggil, <i>mini-map</i> juga muncul pada <i>scene</i> .		

Tabel 13-14 TC-14: Uji unggah video dengan berbagai ukuran

Test Case ID	TC-14			
Nama Test Case	Uji unggah video dengan berbagai ukuran			
Tujuan Test Case	Menguji hingga sebesar apakah ukuran file dapat diunggah			
Use Case Terkait	UC-201, UC-202, UC-203			
Kebutuhan Terkait	KNF-08, KF-11			
Pre-Condition	Membuka halaman untuk mengunggah video			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
<i>Upload file video</i>	1. MP4 10MB 2. MP4 50MB 3. MP4 100MB 4. MP4 190MB	1. Berhasil diunggah 2. Berhasil diunggah 3. Berhasil diunggah 4. Berhasil diunggah	1. Berhasil diunggah 2. Berhasil diunggah 3. Berhasil diunggah 4. Berhasil diunggah	Sukses

Tabel 13-15 TC-15: Uji membuka aplikasi secara bersamaan

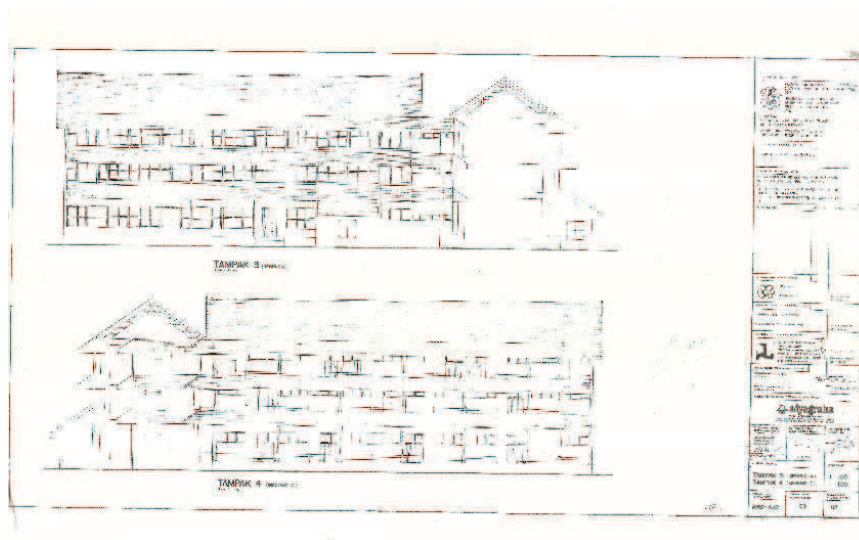
Test Case ID	TC-15			
Nama Test Case	Uji Membuka Aplikasi Secara Bersamaan			
Tujuan Test Case	Menguji apakah aplikasi tetap dapat berjalan apabila ada lebih dari satu orang yang mengakses			
Use Case Terkait	UC-101 s.d. UC-104, UC-109, UC-110			
Kebutuhan Terkait	KNF-05			
Pre-Condition				
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Membuka aplikasi peta 3D dua komputer berbeda dalam satu <i>server</i> .		Aplikasi dapat berjalan dari terbukanya menu utama hingga navigasi	Aplikasi dapat berjalan dan pengguna dapat melakukan navigasi	Sukses

Tabel 13-16 TC-16: Uji membuka video secara bersamaan

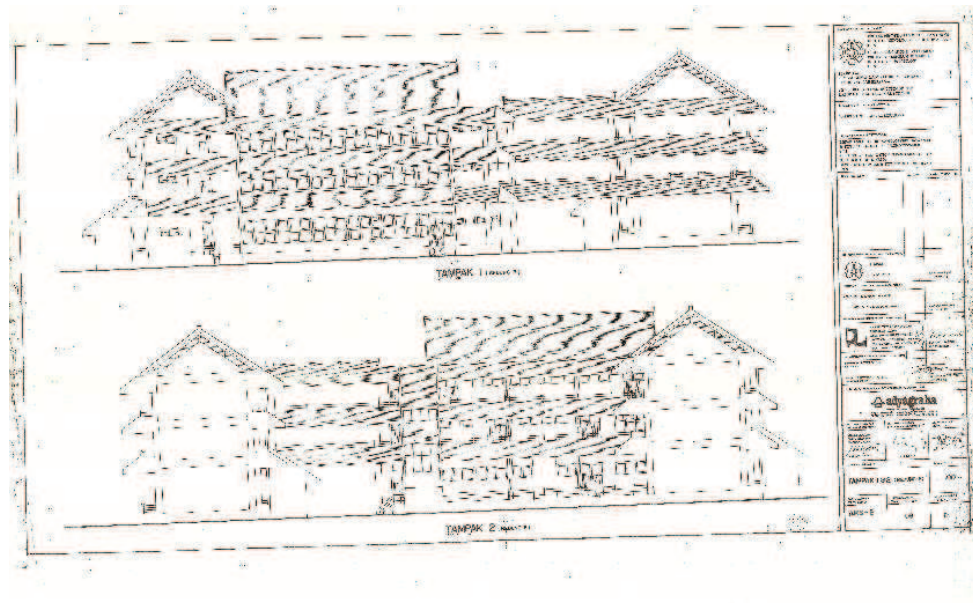
Test Case ID	TC-16			
Nama Test Case	Uji Membuka Video Secara Bersamaan			
Tujuan Test Case	Menguji apakah video tetap dapat dimainkan apabila diakses oleh lebih dari satu orang dalam waktu yang bersamaan.			
Use Case Terkait	UC-105 s.d. 108			
Kebutuhan Terkait	KNF-05			
Pre-Condition	FPS Controller sudah muncul pada masing-masing komputer; Ada koneksi dengan <i>host</i> ; Player berada dalam <i>trigger area</i> ;			
Aksi	Data	Hasil Diharapkan	Hasil Aktual	Sukses/ Gagal
Setiap komputer menekan tombol interkasi secara bersamaan		Video dapat ditampilkan	Video dapat ditampilkan	Sukses

Halaman ini sengaja dikosongkan

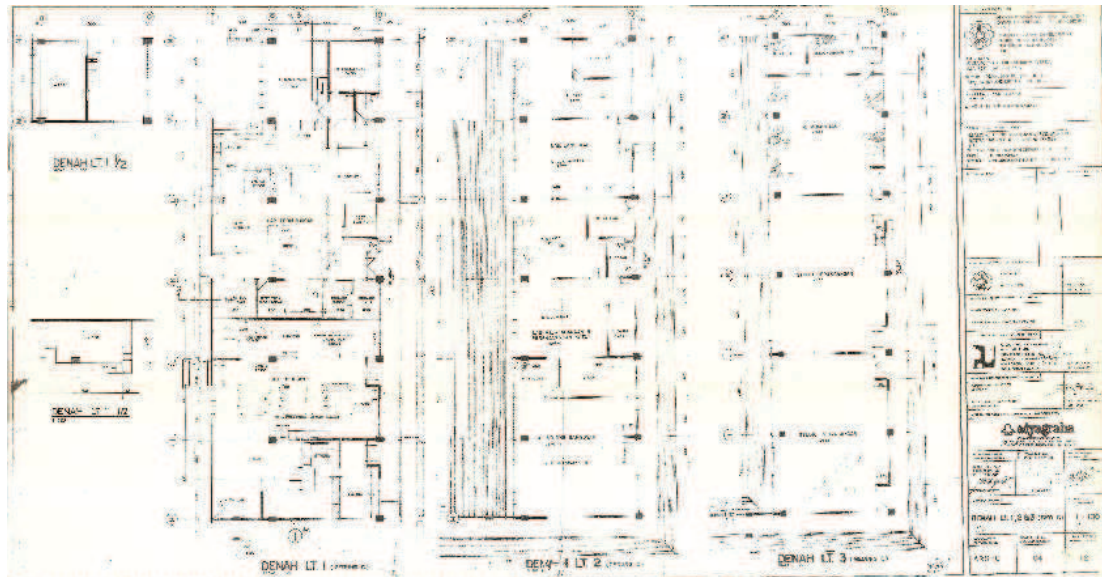
LAMPIRAN E DENAH BANGUNAN



Gambar 14-1 Denah Gedung A Tampak 3 dan Gedung C Tampak 4

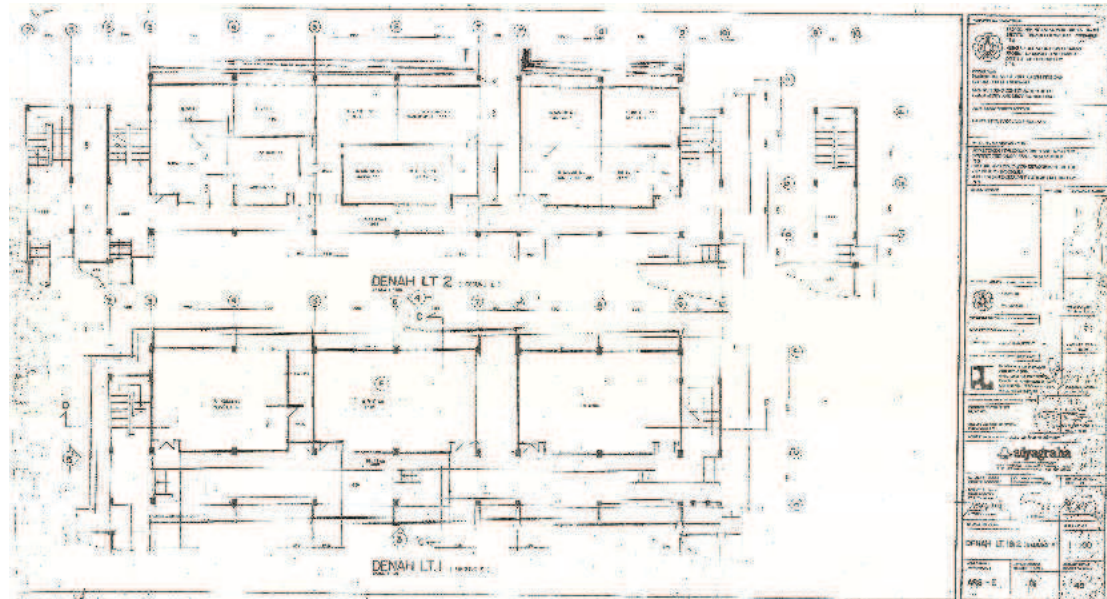


Gambar 14-2 Denah Gedung B Tampak 1 dan 2



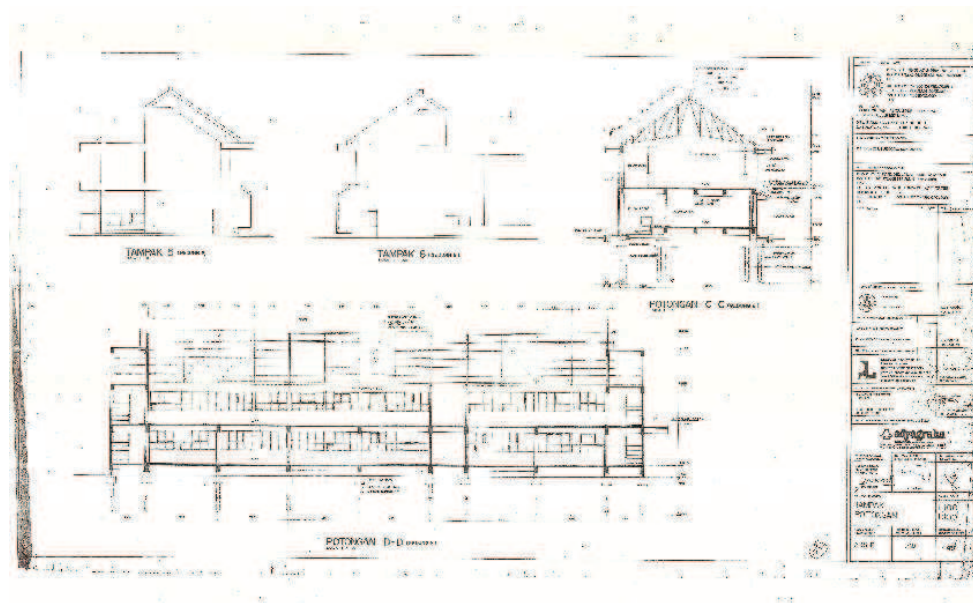
Gambar 14-3 Denah Gedung C lantai 1, 2, dan 3

E-4

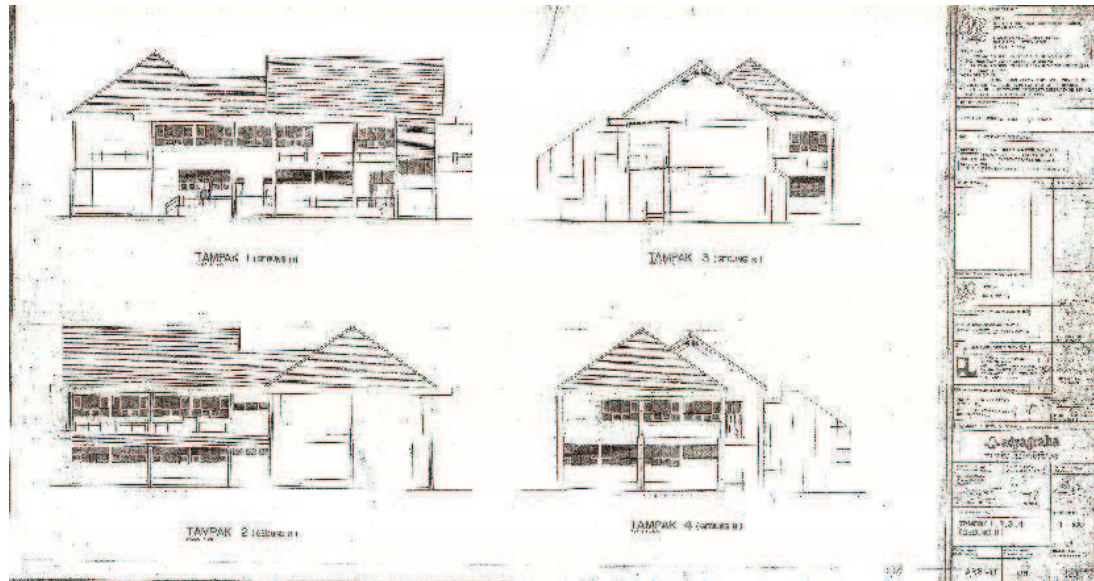


Gambar 14-4 Denah Gedung E lantai 1 dan 2

Gambar 14-5 Denah Gedung E tampak 3 dan 4

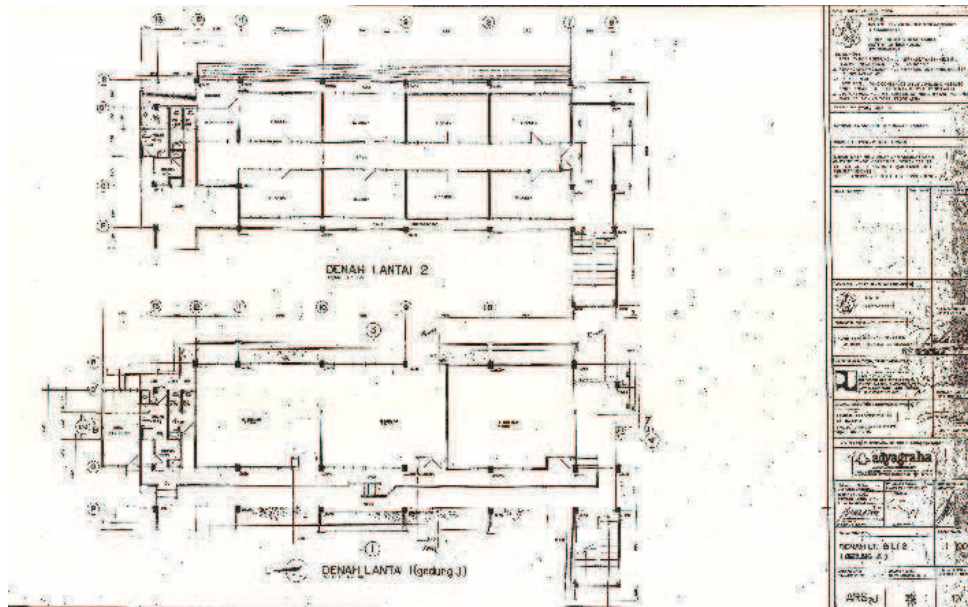


Gambar 14-6 Denah Gedung E Tampak Potongan

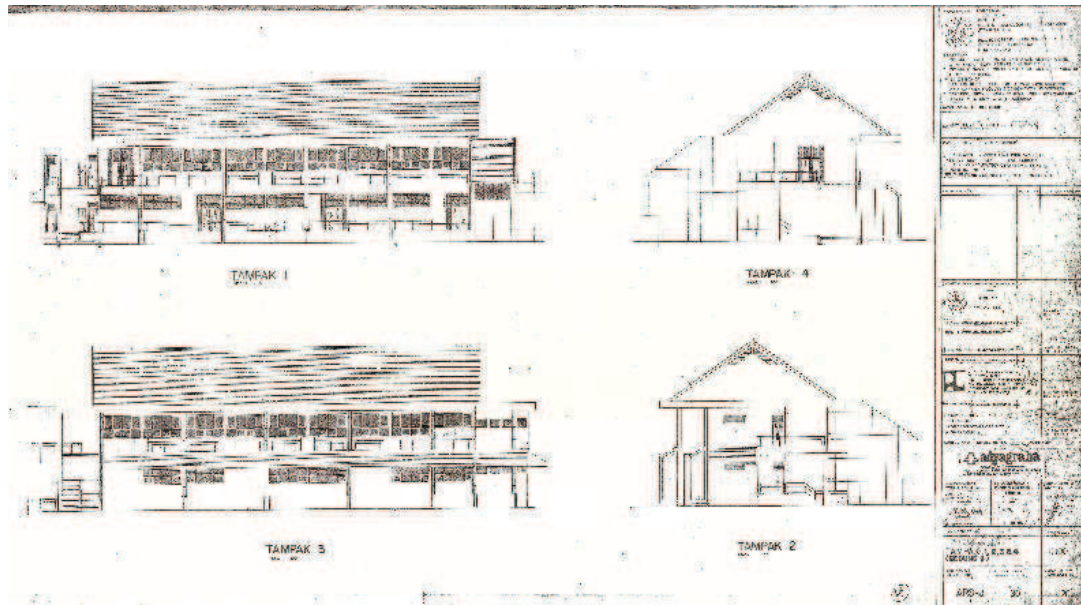


Gambar 14-8 Denah Gedung H Tampak 1, 2, 3, dan 4

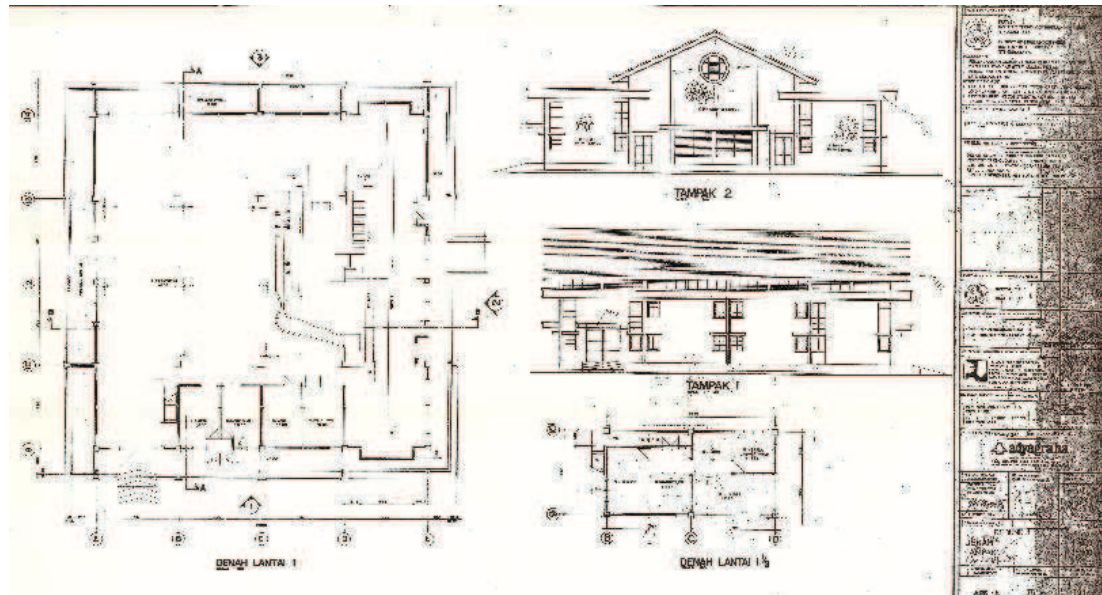
Gambar 14-10 Denah Gedung I Tampak 1, 2, 3, dan 4



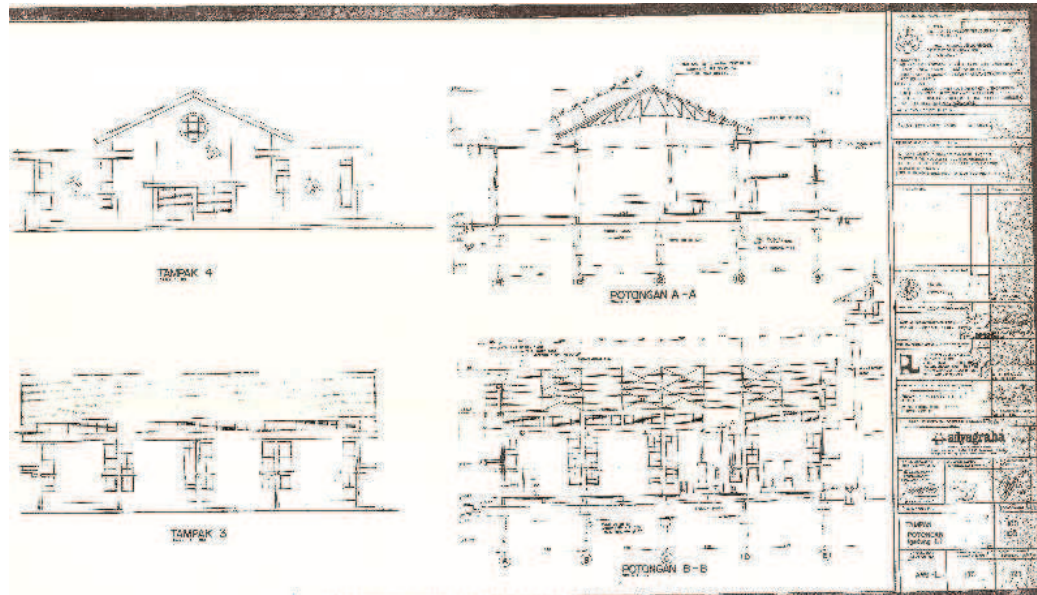
Gambar 14-11 Denah Gedung J Lantai 1 dan 2



Gambar 14-12 Denah Gedung J Tampak 1, 2, 3, dan 4



Gambar 14-13 Denah Gedung L Lantai 1, Lantai 1 1/2, Tampak 1, dan Tampak 2



Gambar 14-14 Denah Gedung L Tampak Potongan